

Návrh a implementace systému sběru a zpracování dat z přenosné platformy sloužící pro identifikaci závodníků

Design and Implementation of Complex System for Data Collection and
Processing from Portable Platforms for Identifying Competitors

Bc. Kamila Lišková

Diplomová práce

Vedoucí práce: Ing. Radovan Hájovský, Ph.D.

Ostrava, 2021

Abstrakt

Tato diplomová práce popisuje systém sloužící pro identifikaci závodníků a vyhodnocení výsledků orientačního závodu. Systém se skládá z kontrolních zařízení a aplikace pro správu a vizualizaci dat. Úkolem kontrolních zařízení je vytvořit záznam o průchodu účastníka přes stanoviště. Součástí těchto zařízení je modul pro komunikaci přes LoRaWAN a je možné vidět data v reálném čase. Tato zařízení jsou napájena z baterií, z toho důvodu je provedena i analýza energetické spotřeby. Byla vytvořena webová aplikace pro administraci a vizualizaci dat. Administrace je uživatelsky přívětivá a je možné ji dynamicky použít na více orientačních závodů. Vizualizace je vytvořena pomocí jazyka R.

Klíčová slova

IoT technologie; LoRaWAN; RFID; R; Shiny; SQL; Orientační závod; Raspberry Pi; Vizualizace

Abstract

This diploma thesis describes a system designed and used to identify competitors and evaluate the results of the orienteering race. The system consists of control devices and an application for data management and its visualization. The task of the control devices is to create a record of the competitor's passage through the site. The devices include a module for communication via LoRaWan which enables to see data in real time. These devices are powered by batteries therefore an analysis of energy consumption is performed. An web application for data administration and visualization has been created as well. The visualization is user-friendly and can be used dynamically for multiple orienteering races. The visualization has been created using the R language.

Keywords

IoT technologies; LoRaWAN; RFID; R; Shiny; SQL; Orienteering race; Raspberry Pi; Visualization

Obsah

Seznam použitých symbolů a zkratk	5
Seznam obrázků	6
Seznam tabulek	8
1 Úvod	9
2 Seznámení se s IoT technologií LoRa	11
2.1 Síť LoRa a LoRaWAN	13
2.2 Příjem a zpracování dat	22
2.3 Vizualizace dat pomocí jazyka R	25
3 Návrh a realizace infrastruktury pro sběr dat z kontrolních stanovišť s využitím bezdrátové technologie LoRa	29
3.1 Požadavky na kontrolní zařízení	30
3.2 Použitý hardware v kontrolních zařízeních	30
3.3 Popis Firmware	37
3.4 Pokrytí hracího pole sítí LoRaWAN	43
4 Analýza energetické náročnosti jednotlivých komponent infrastruktury	44
4.1 Měření odběru jednotlivých komponent	45
4.2 Měření odběru celého zařízení	45
4.3 Odhad výdrže na baterii	48
5 Návrh a implementace SW aplikace pro zpracování a vizualizaci dat o závodnících	50
5.1 Pravidla a průběh hry	51
5.2 Použité technologie	52
5.3 Sběr a administrace dat	55
5.4 Zpracování dat a výpočet výsledků	60

5.5	Vizualizace dat	63
6	Zhodnocení dosažených výsledků	67
7	Závěr	68
	Literatura	70
	Přílohy	72
A	Obrázky	73
A.1	Ukázka mapy herního území	74
A.2	Schéma zapojení	75
A.3	Proudový odběr simulace provozu	76
B	Testování	77
C	Zdrojové kódy	83

Seznam použitých zkratek a symbolů

ABP	– Authentication By Personalisation
BW	– Bandwidth
CF	– Carrier Frequency
CR	– Coding Rate
CRC	– Cyclic redundancy check
CSS	– Chirp Spread Spectrum
CRAN	– Comprehensive R Archive Network
FSK	– Frequency Shift Keying
IoT	– Internet of Things
IP	– Internet Protocol
LED	– Light-Emitting Diode
LoRa	– Long Range
LoRaWAN	– Long Range Wide Area Network
LPWAN	– Low Power Wide Area Network
MAC	– Medium Access Control
MQTT	– Message Queuing Telemetry Transport
OTAA	– Over The Air Activation
PCB	– Printed Circuit Board
RF	– Radio Frequency
RFID	– Radio Frequency Identification
RTC	– Real-time clock
SF	– Spreading Factor
SNR	– Signal to Noise Ratio
SQL	– Structured Query Language
TCP/IP	– Transmission Control Protocol / Internet Protocol
TTN	– The Things Network
USB	– Universal Serial Bus
RPi	– Raspberry Pi

Seznam obrázků

2.1	Počet připojených IoT zařízení	12
2.2	CSS – Průběh nosného a modulovaného signálu	14
2.3	Vztah mezi spreading faktorem a vysílacím časem pro modulaci LoRa	15
2.4	Struktura rámce LoRa	16
2.5	Síť LoRaWAN	17
2.6	LoRaWAN Třída A: Časování vyslání a příjmu koncového zařízení	18
2.7	Aktivace modulu LoRaWAN	20
2.8	The Things Network síťová architektura	23
2.9	MQTT architektura	24
2.10	Schéma Shiny aplikace	26
3.1	Zjednodušené blokové schéma celého systému	29
3.2	Blokové schéma kontrolních zařízení	30
3.3	Zapojení – tlačítko	33
3.4	Zapojení – LED a bzučák	34
3.5	Zapojení – měření napětí baterie	34
3.6	Deska plošných spojů	35
3.7	Finální podoba zařízení	36
3.8	Diagram RFID komunikace – vlevo celý průběh, vpravo jen zápis dat na čip	39
3.9	Datový formát RFID	40
3.10	Datový formát LoRa	41
3.11	Mapa pokrytí oblasti s herním polem	43
4.1	Proudový odběr obsluhy jednoho závodníka	46
5.1	První stránka – přihlášení	55
5.2	Zobrazení údajů – stanoviště	56
5.3	Editace údajů	56
5.4	Sběr dat – zobrazení na mobilním telefonu	57

5.5	Komunikace s RFID čipy	58
5.6	Schéma toku dat LoRa	59
5.7	Schéma databáze	60
5.8	Zpracované výsledky	62
5.9	Vizualizace – online závod	63
5.10	Vizualizace – box s hodnotou	64
5.11	Vizualizace – navštívené stanoviště	65
5.12	Vizualizace – účastníci na stanovišti	65
5.13	Vizualizace – trasy závodníků	66
A.1	Ukázka mapy herního území	74
A.2	Schéma zapojení	75
A.3	Proudový odběr simulace provozu	76

Seznam tabulek

2.1	Přehled typů MAC zpráv	19
3.1	Základní parametry ATmega328P	31
3.2	Základní parametry RN2483	31
3.3	Základní parametry DS3231	32
3.4	Základní parametry MF RC522	32
3.5	Základní AT příkazy pro komunikaci s LoRa modulem	40
4.1	Proudový odběr jednotlivých modulů	45
4.2	Porovnání různých scénářů použití zařízení	47
4.3	Výdrž baterie v různých scénářích použití	48
5.1	Přehled bodování	51

Kapitola 1

Úvod

Internet věcí, neboli Internet of Things, označovaný zkratkou IoT, je prudce rostoucí obor informačních technologií, jehož hlavním cílem je připojení mnoha zařízení do jednotné sítě, což umožní centralizovaný vzdálený dohled. Popularita systému IoT roste kvůli neustále se rozrůstajícímu bezdrátovému pokrytí a z důvodu cenově dostupných elektronických komponentů.

Jedna ze zajímavých IoT technologií je LoRa. Je to bezdrátový telekomunikační systém s velkým dosahem, nízkou spotřebou, avšak s nízkou přenosovou rychlostí. Koncová zařízení komunikují bezdrátově s bránou, která je připojená k internetu. Tyto brány fungují jako transportní mosty a přenášejí zprávy mezi koncovými zařízeními (senzory) a centrálním síťovým serverem.

Cílem této diplomové práce je navrhnout a realizovat systém sloužící pro identifikaci závodníků a vyhodnocení výsledků orientačního závodu. Technologie LoRa je v této práci použita pro lepší přehled o závodu. Data ze zařízení jsou posílána přes internet až na server, kde jsou uložena do databáze a následně vizualizována. Díky tomu budou mít organizátoři lepší přehled o aktuálním stavu závodu.

Hlavním požadavkem na navrhovaný kontrolní systém je umožnit použití osobě bez znalostí použitých technologií. Práce je zaměřena na dostupné zpracování dat před a po závodu. Zejména načítání dat z RFID čipů.

Práce je rozdělena na teoretickou část, zabývající se IoT technologií, a praktickou část, týkající se návrhu a realizaci systému. V teoretické části je popsána technologie LoRa a možnost vizualizace dat pomocí jazyka R.

Praktická část je rozdělena na kapitoly popisující dílčí úkoly praktické části. Ve třetí kapitole je popsán návrh a realizace zařízení pro sběr dat o účastnících. Úkolem těchto zařízení je vytvořit záznam o průchodu účastníka přes kontrolní stanoviště.

Čtvrtá kapitola je zaměřena na analýzu energetické náročnosti navrhovaných zařízení. Je změřena náročnost jednotlivých modulů a zařízení jako celku. Z naměřených dat byl stanoven odhad výdrže zařízení na baterii. Kapitola obsahuje i popis činnosti zařízení pomocí grafického znázornění proudového odběru.

Pátá kapitola se zabývá návrhem a implementací aplikace pro zpracování a vizualizaci dat. Pro každý závod je založena nová databáze. Pomocí aplikace je možno editovat záznamy v databázi, vyhodnotit výsledky závodu a vizualizovat všechna dostupná data o závodu. Je možno založit nový závod, zavést data o závodnících a stanovištích, sledovat živý průběh závodu a na konci ihned vytvořit výsledkovou listinu.

V šesté kapitole je popsáno testování navrhovaného systému. Testování proběhlo ve dvou hlavních krocích. V domácích podmínkách byly otestovány jednotlivé funkce a v terénu proběhl zkušební závod.

Kapitola 2

Seznámení se s IoT technologií LoRa

V dnešní době se je čím dál častěji možno setkat s pojmem IoT (Internet Of Things). Představa, že každé fyzické zařízení dokáže komunikovat a zaznamenávat data, přináší velkou výhodu. Myšlenka chytrého města vznikla již dávno, nicméně díky této technologii může být její realizace mnohem jednodušší a cenově přívětivější.

Myšlenka sítě sestávající z inteligentních zařízení začala být diskutována již v roce 1982, kdy na univerzitě v Pensylvánii připojili k internetu automat na nealkoholické nápoje. Díky tomu obsluha byla schopna na dálku zjistit informace o množství založených nápojů, nebo zda už jsou nově založené nápoje dostatečně vychlazené. V roce 1994 popsala Reza Raji základní pojetí IoT v časopise IEEE Spectrum jako „přesun malých datových paketů do velké sady uzlů, které integrují a automatizují vše od domácích spotřebičů po celé továrny“. [1]

Jednou z nejvíce vypovídacích definic uvedl v roce 2010 McKinsey: Internet věcí = „Senzory a akční členy zabudované do fyzických objektů jsou propojeny prostřednictvím kabelových a bezdrátových sítí, často využívajících stejný internetový protokol (IP), který spojuje internet.“ [2]

Během posledních několika let se internet věcí změnil ve vyspělou technologii s aplikacemi v průmyslovém i spotřebitelském prostředí. Revoluce IoT slibuje, že vytvoří ambientní inteligenci propojení všech zařízení a objektů kolem nás a umožní jim přizpůsobit se a chovat se podle potřeb lidí. Výsledkem je široká škála zabudovaných hardwarových zařízení, různé komunikační protokoly s důrazem na účinnost a bezpečnost a spousta softwarových platforem určených ke správě všech těchto zdrojů. [3]

Prakticky vzato, je IoT zařízení vše, co je připojeno k síti a je možno s tím komunikovat bez přímého fyzického kontaktu. Využití IoT je opravdu široké. Ať je to průmysl, kde umožňuje poměrně efektivně udržovat kontrolu nad jednotlivými stroji či zařízeními, ale také například nad stavem kávovaru v kantýně. Uplatnění lze najít i v domácnosti, například inteligentní pračka, lednička, či celá inteligentní domácnost není v dnešní době už nic neobvyklého. IoT zařízení rovněž nachází uplatnění v oblastech životního prostředí, kde je možné snímat a zaznamenávat kvalitu ovzduší,

četnost srážek, stavu hladiny vodních toků, a tím například mohou pomoci k minimalizaci škod zapříčiněnými živelnými pohromami.

Zásadní rozdíl mezi „internetem“ a „internetem věcí“ (IoT) je v tom, že pro IoT zařízení je k dispozici „méně všeho“ – méně paměti, méně výpočetního výkonu, menší šířka pásma a samozřejmě méně dostupné energie. Je to hlavně proto, že „věci“ jsou poháněny bateriemi a očekává se jejich velký počet (odhad z roku 2018 je uveden na obrázku 2.1). Následně díky tomu jsou i nižší pořizovací náklady. Tato snaha „dělat více s méně“ vede k omezení použitelnosti tradičních celulárních sítí i technologií, jako je třeba WiFi nebo i GSM, kvůli požadavkům na energii a škálovatelnost. [4]



Obrázek 2.1: Počet připojených IoT zařízení od roku 2015 do 2025 (v bilionech) [5]

Možností připojení zařízení do sítě je mnoho. Některé technologie vyžadují, aby zařízení bylo neustále připojeno do sítě a naopak některá zařízení jsou připojena do sítě pouze v pravidelných intervalech, či při potřebě odeslat data. Zařízení se také liší ve směru komunikace. Existují zařízení, která mají pouze jednosměrný kanál komunikace, tzn. pouze odesílají data, ale existují také zařízení, která jsou schopna kromě odesílání i data přijímat, a tím například upravovat svou činnost na základě vnějších požadavků. Další důležitý parametr pro výběr použité technologie je velikost přenášených dat.

Mezi často používané technologie v IoT patří skupina technologií označující se LPWAN (Low Power Wide Area Networks). Jejich obrovská výhoda spočívá v nízké spotřebě energie a velkému rádiovému dosahu. Tyto výhody jsou zároveň požadavkem pro IoT sítě.

Low Power Wide Area Networks

Volným překladem se jedná o nízkoenergetickou síť pro rozsáhlé území. Jedná se o spojení dvou charakteristik, kterými jsou malá spotřeba energie (delší životnost baterie) a velký dosah signálu. Mezi nevýhody patří nízká rychlost dat a mnohdy i omezené množství přenášených dat. Technologie

LPWAN může fungovat na licencovaném i nelicencovaném spektru a zahrnuje proprietární, alianční nebo otevřené standardní možnosti.

Sítě LPWAN typicky pracují na rádiové frekvenci 800–900 MHz. Nižší frekvence má hned několik výhod – nižší energetická náročnost, lepší dosah a snazší prostup signálu překážkami. Nicméně snižování frekvence má své praktické limity ve velikosti antény (čím nižší frekvencí, tím větší je délka vlny a tedy je potřebná větší anténa).

Pojmem LPWAN je označena skupina technologií, které splňuje potřebné vlastnosti. Dnes existuje celá řada LPWAN sítí, ale typicky se bavíme o třech hlavních představitelích: NB-IoT, LoRaWAN a Sigfox. V této práci je použita technologie LoRaWAN hlavně z důvodu nulových poplatků za provoz. [6] Následuje její popis.

2.1 Síť LoRa a LoRaWAN

LoRa je zkratka pro „Long Range“. Jak je zjevné z názvu, nejdůležitější vlastností LoRa je komunikace na velké vzdálenosti. I když jsou termíny LoRa a LoRaWAN používány zaměnitelně, odkazují však na dva různé pojmy. Pojem LoRa ve skutečnosti odpovídá fyzické vrstvě, používající modulační techniku Chirp Spread Spectrum (CSS), která je používána pro vytvoření telekomunikačního spojení na velkou vzdálenost. O vývoj se stará společnost Semtech. A druhý termín LoRaWAN definuje komunikační protokol a síťovou architekturu systému. Je to protokol, jehož cílem je zabezpečení vrstvy MAC (Medium Access Control) v síti LoRa. Je navržen LoRa Aliancí, což je spolek, který sdružuje více než 400 členů (telekomunikační operátoři, výrobci polovodičů, společnosti zabývající se digitálním zabezpečením, výrobci hardwaru, dodavatelé sítí atd.). Zatímco modulace LoRa je proprietární, LoRaWAN je otevřený standard vyvíjen Aliancí LoRa.

2.1.1 Fyzická vrstva LoRa

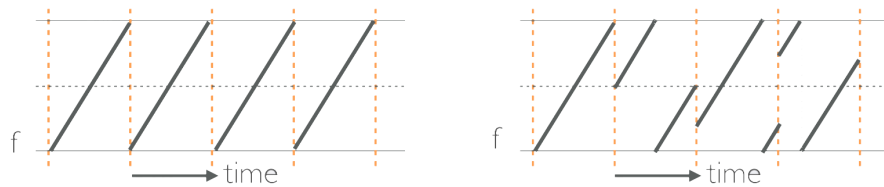
Jak už bylo zmíněno, modulace LoRa patří mezi proprietární modulace, tedy její přesná hardwarová implementace není veřejně přístupná. O vývoj se postarala firma Semtech. Pro vysílání jsou využívána bezlicenční frekvenční pásma ISM. Ta se v jednotlivých částech světa liší dle regionálních regulí, avšak nejčastěji LoRa operuje na těchto nižších kmitočtech (EU: 868 MHz a 433 MHz, USA: 915 MHz a 433 MHz). Nabízí různé konfigurace (přenosové rychlosti, rozsah přenosu, spotřeba energie a odolnost vůči šumu) podle výběru parametrů. Pásma ISM (industrial, scientific and medical) jsou pásma pro rádiové vysílání v oborech průmyslovém, vědeckém a zdravotnickém. Jsou to pásma volná, což znamená, že je v nich při použití homologovaného (schváleného) zařízení dovolen provoz bez licenčních poplatků, avšak bez garance proti rušení.

LoRa modulace

LoRa technologie je založená na modulační technice Chirp Spread Spectrum (CSS) používané ve vojenské a vesmírné komunikaci, především z důvodu dlouhého dosahu a odolnosti vůči rušení. Při

použití CSS modulace je generován takzvaný chirp signál, jehož pulzy se dají označit jako sinusové pulzy, které mění svou frekvenci. [7]

Signály chirpů mají konstantní amplitudu a zabírají celou šířku pásma. Frekvence se mění lineárně s časem. Frekvence se buď zvyšuje od spodní po horní hranice pásma. Takový signál je označen jako chirp-up. Nebo v opačném případě se frekvence snižuje - označován chirp-down. Po dosažení horní hranice se frekvence vrací na spodní hranici a proces se opakuje. Na tento signál je následně přerušováním (chirp) modulována přenášená informace. Je znázorněno na obr. 2.2. Na levé straně se nachází nosný signál v podobě chirp-up pulzů a na pravé straně je zobrazen modulovaný signál.



Obrázek 2.2: CSS – Průběh nosného a modulovaného signálu [8]

Díky jednoduché linearitě chirp pulzů jsou frekvenční rozdíly mezi vysílačem a přijímačem shodné s časovými rozdíly, proto je možné je jednoduše eliminovat. To zvyšuje odolnost vůči Dopplerovu jevu a snižuje nároky na zdroj kmitočtu. Navíc jeden kanál je možné využívat současně pro několik vysílačů a umožňuje komunikaci při pohybu zařízení.

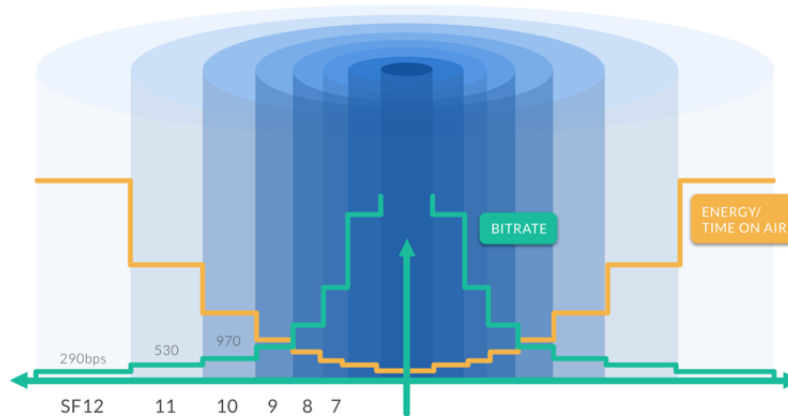
Frekvenční rozdíl může být až 20 % šířky pásma bez vlivu na schopnost dekodovat signál, což pomáhá snížit cenu LoRa modulů – není nutné implementovat vysoce přesné krystaly, které zvyšují koncovou cenu. Není ani nutné aplikovat komplexní algoritmy, což snižuje energetickou náročnost. [9] Modulaci je možné upravit pomocí několika parametrů.

Lora parametry

Modulace má několik volitelných možností pro rozptřeni (Spreading Factor – SF) a šířku pásma (Bandwidth – BW), pevně stanovených pro každou datovou rychlost a kódový poměr (Coding Rate, CR). Pro stacionární aplikace se doporučuje zapnout ADR (Automatic Data Rate), která nastaví automaticky nejvyšší možnou přenosovou rychlost na základě naměřeného signálu bránu. [10]

Spreading Factor neboli faktor šíření, určuje poměr mezi rychlostí symbolu a rychlostí čipu. Jinak řečeno – jaké množství dat je vysláno za jednotku času. Čím vyšší je spreading faktor, tím se zvyšuje citlivost a rozsah. Také se zvyšuje poměr signálu a šumu (SNR) i čas vysílání paketu, což znamená menší množství dat za sekundu. Delší vysílací čas navíc znamená, že zařízení je v provozu déle a spotřebovává více energie. Spreading faktor lze zvolit od 6 do 12. Výsledná hodnota pak odpovídá mocninám čísla 2. Jednotlivé SF jsou vzájemně ortogonální, tedy mohou být použity na

stejné frekvenci, aniž by docházelo k interferencím. [11] Na obrázku 2.3 je možné vidět vztah mezi spreading faktorem a přenosovou rychlostí.



Obrázek 2.3: Vztah mezi spreading faktorem a přenosovou rychlostí pro modulaci LoRa [12]

Technologie LoRa využívá ke komunikaci 8 kanálů. Z toho by logicky plynulo, že v jeden časový okamžik je brána schopna detekovat pouze 8 zpráv – na každém kanálu jednu. To však zcela obecně neplatí, protože na každém kanálu je kontinuálně prováděna signálová analýza, která je za specifických podmínek schopna detekovat až 4 zprávy současně. Specifické podmínky v tomto případě spočívají v tom, že každá detekovaná zpráva je kódována jinou úrovní spreading faktoru. Právě z toho důvodu je také výhodné, aby řízení radiových parametrů pro komunikaci s čidly bylo svěřeno network serveru, který zajistí co nejrovnoměrnější distribuci spreading faktorů mezi jednotlivá čidla, a navýší tím propustnost sítě. [10] [13]

Bandwidth (šířka pásma) ovlivňuje především datovou rychlost, citlivost přijímače a obsazenou část spektra. Při jejím nastavování je potřeba se řídit regionálními normami definující parametry vysílání na daných frekvencích. [10]

Coding Rate (kódový poměr) definuje rozšíření délky dat oproti jejich původní délce. Rozšíření je z důvodu zvýšení ochrany před možnými chybami v přenosu kanálem. Využívá se cyklického kódu (CRC), který za poslanou zprávu přidá vlastní informaci. Dodatečná informace se použije pro detekci chyb, a pokud to bude možné, i následnou korekci. LoRa Alliance definuje čtyři kódové faktory. Čím vyšší je jeho hodnota, tím více chyb bude možno při zpětném dekódování opravit. [10]

Struktura zpráv fyzické vrstvy

Podobu zpráv na fyzické vrstvě definuje použitá technologie LoRa. Dle směru vysílání se dělí na uplink a downlink. Formát těchto zpráv je podobný. **Uplink** jsou zprávy vyslány od koncových zařízení směrem k řídicímu serveru. A **downlink** jsou označovány zprávy putující od řídicího serveru

ke koncovým zařízením. Od uplinku se liší pouze v tom, že datová část neobsahuje kontrolní součet CRC. Důvodem je zajistit co možná nejkratší délku paketu.

Ačkoli lze použít modulaci LoRa k přenosu libovolných rámců, je ve vysílačích a přijímačích vyrobených firmou Semtech specifikován a implementován fyzický formát rámce. Bandwidth a spreading faktor jsou pro rámec konstantní. Na obrázku 2.4 je vidět strukturu rámce na fyzické vrstvě.

Na začátku rámce LoRa je **Preamble**. Jejím úkolem je především jednoznačně umožnit detekci a synchronizaci na příchozí paket. Preamble obsahuje synchronizační slovo – jedná se o jednobajtovou hodnotu, která slouží k rozlišení sítí LoRa, které používají stejná frekvenční pásma. [4]

Po preamble následuje volitelná **Header** (hlavička). Obsahuje velikost Payload (užitečných dat) (v bajtech), použitý coding rate a to, zda je na konci rámce 16bitový CRC. Hlavička také obsahuje CRC. To umožňuje přijímači zahodit pakety s neplatnou hlavičkou. Velikost užitečných přenášených dat je uložena pomocí jednoho bajtu, což omezuje velikost dat na 255 bajtů. Hlavička je volitelná, aby bylo možné jej deaktivovat v situacích, kdy to není nutné, například když je předem známa délka dat, rychlost kódování a přítomnost CRC. [4]

Dále následuje **Payload** (užitečná data) a CRC těchto dat. Jedná se o jakákoliv data až o maximálním rozsahu 256 B a díky nastavitelnému kódovému poměru si lze vybrat robustnost zabezpečení proti chybám. Volitelně lze doplnit uživatelská data na konci o kontrolní součet.



Obrázek 2.4: Struktura rámce LoRa [14]

2.1.2 Protokol LoRaWAN

Protokol LoRaWAN je optimalizován pro zařízení napájena pomocí baterií - mobilní i pevně umístěna. Komunikace je obousměrná a je rozprostřena mezi různými frekvenčními kanály a přenosové rychlosti, přičemž zde nedochází k interferenci. Rychlost přenosu se pohybuje od 0,3 kb/s do 50 kb/s.

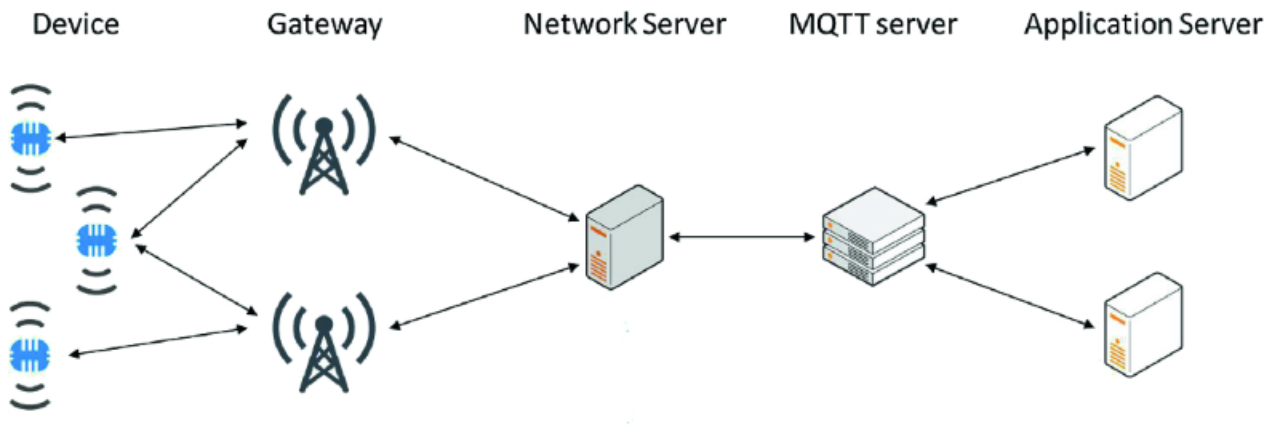
LoRaWAN je protokol MAC vrstvy vytvořený pro použití fyzické vrstvy LoRa. Protokol poskytuje mechanismus řízení přístupu ke vrstvě MAC, umožňuje tak koncovým zařízením komunikovat s bránou za použití LoRa modulace.

Tento protokol využívá metodu přístupu k médiu typu Aloha. V síti mesh nebo celulární síti se musí zařízení probouzet kvůli synchronizaci se sítí. Tato synchronizace spotřebovává energii a snižuje životnost baterie. Komunikace v LoRaWAN síti probíhá asynchronně. Koncová zařízení komunikují pouze pokud mají data k odeslání, ať už data jsou odesílána nepravidelně nebo jsou periodicky plánována.

V následujícím textu je popsána architektura sítě, způsob komunikace a struktura zpráv.

Architektura sítě

Strukturu sítě LoRaWAN lze rozdělit do několika částí. Nejvíce početnou skupinu tvoří uživatelská koncová zařízení (End-devices), následována velkým množstvím bran (Gateways). Další a nejdůležitější část představuje řídicí server a poslední skupinou jsou aplikační servery. Sít LoRaWAN odpovídá topologii hvězda. Obrázek 2.5 ukazuje komponenty typické pro tuto síťovou infrastrukturu. Následuje popis jednotlivých komponent.



Obrázek 2.5: Sít LoRaWAN (zjednodušené zobrazení) [15]

Koncová zařízení (end-Devices) – nebo (LoRa) uzly. Slouží k odesílání malého množství dat při nízkých frekvencích na velké vzdálenosti. Koncové zařízení většinou obsahují různé senzory (detektory kouře, teploměry, hlukoměry či senzory kvality ovzduší) a několikrát za den jsou naměřené hodnoty posílány na server. Zařízení je většinou napájeno z baterie, proto je jejich optimalizace zaměřená především na spotřebu a dlouhou životnost. Může se jednat o různé aplikace, jako je inteligentní město, inteligentní budova, automatizace továren, automatizace farem nebo logistika. [8]

Brána (Gateway) – neboli základní stanice. V hvězdicové topologii představuje ústřední prvek, který přijímá datové pakety od všech zařízení v okolí. Tato data jsou pak přeposílána na centrální síťový server (Network Server), zpravidla po TCP/IP (Internet Protokol) síti – Ethernet, WiFi, 3G/LTE. Ke každému přijatému paketu z koncového zařízení jsou bránou přidány informace (kvalita příjmu, čas atd.). Sít lze jednoduše rozšířit o další brány a umožnit zvýšení datové rychlosti koncových zařízení. [8]

Síťový server (Network Server) – centrální prvek, který hraje důležitou roli spolu s **MQTT serverem**. Komunikuje s jednotlivými branami a v případě, že jsou stejná data přijata od více bran, tato data agreguje. Provádí úkony, jako jsou detekce redundantních paketů, bezpečnostní kontrolu a plánování downlink paketů. Musí splňovat LoRaWAN specifikaci a má řídicí úlohu (např. může měnit vysílací výkon na koncovém zařízení).

Aplikační server (Application Server) – operuje nad samotnými daty. Data mohou být uložena do databáze a následně mohou být zdrojem statistik a grafů.

Stojí za zmínku, že v síti LoRaWAN nejsou uzly pevně připojeny k žádné konkrétní bráně a paket odeslaný uzlem může být přijímán více branami v dosahu komunikace uzlu. Specifické předávací postupy při přemístění zařízení tedy nejsou nutné, což eliminuje signalizaci související s předáním. To přináší energetické úspory v uzlech a zvyšuje pravděpodobnost úspěšného doručení paketu. [16]

Způsoby komunikace - Třídy

Protokol LoRaWAN definuje tři způsoby (třídy), jak koncové zařízení komunikuje s bránou. Hlavní rozdíl mezi nimi je ve spotřebě energie a jak často zařízení naslouchá zprávě vyslanou bránou (downlink). Třída A má nejnižší požadavky na spotřebu energii. Zařízení třídy B periodicky aktivují svou rádiovou přijímací část. Zařízení třídy C je vhodné použít tam, kde je potřeba stále otevřená komunikace mezi koncovými zařízeními a branami, což výrazně zvyšuje spotřebu energie. [13] [17] [18]

Třída A: Jedná se o základní třídu, kterou musí mít implementována všechna koncová zařízení. Její použití je vhodné v situacích, kdy je k dispozici jen napájení z baterie a je požadavek hlavně na posílání dat ze senzoru (koncového zařízení). Umožňuje však i obousměrnou komunikaci, ale v omezeném režimu. Přijmout zprávu vyslanou z brány do zařízení je možné jen krátce po odeslání zprávy ze zařízení. Po uplink přenosu následují dvě krátká downlink přijímací okna (znázorněné na obrázku č. 2.6).



Obrázek 2.6: LoRaWAN Třída A: Časování vyslání a příjmu koncového zařízení [8]

Zařízení třídy A tedy vždy zahajuje komunikaci, a to na základě nějaké události (např. detekce kouře u kouřového čidla), nebo v závislosti na intervalu časovače uvnitř čidla. Po odeslání zprávy je zařízení schopno ve dvou intervalech o celkové délce pár sekund přijmout zprávu ze serveru, na základě které pak může vyvolat nějakou akci (např. sepnout relé, přenastavit interval odesílání zpráv apod.). Čidla třídy A jsou tedy po většinu času v režimu spánku a díky tomu spotřebovávají jen malé množství energie.

Prodleva RX delay 1 nabývá jednotek sekund a je nastavena bránou, ke které je koncové zařízení připojeno. Druhé přijímací okno se otevírá sekundu po prvním, avšak pouze v případě, že nebyla přijata žádná zpráva. Délka obou oken musí být dostatečně dlouhá pro příjem a dekodování hlavičky LoRa paketu.

Třída B: V mnoha uživatelských aplikacích je vyžadována častější komunikace. V takových případech je vhodné použít zařízení třídy B. Mezitím co u třídy A je nutné každý příjem dat iniciovat vlastním odesláním zprávy, u třídy B jsou otevírána další přijímací okna v synchronizovaných časových intervalech. Takže v pravidelném časovém intervalu aktivují svoji přijímací radiovou část, aniž by došlo k vysílání zprávy (což již představuje pro zařízení energeticky nejnáročnější operaci).

Pomocí synchronizačních impulsů, které se šíří sítí, se interní časovače zařízení synchronizují s časem v síti. Interní časovač lze pak nastavit tak, aby se zařízení aktivovalo pro příjem zprávy v intervalu, který vyžaduje daná aplikace. Interval můžeme nastavit na příklad na jednu minutu, pokud chceme, aby zařízení zareagovalo na downlink zprávu nejdéle do jedné minuty.

Třída C: Zařízení třídy C mohou být použita v případech, kdy je požadavek na okamžitou odezvu na downlink zprávy a není vyžadována nízká spotřeba. Zařízení třídy C neustále naslouchají příchozím zprávám a jejich přijímací radiová část je tedy stále aktivní, což způsobuje jejich vyšší energetickou náročnost. Typicky se jedná o akční členy, které nejsou napájené z baterie, a slouží na příklad k ovládání strojů, spínání relé apod.

Typy zpráv

LoRaWAN rozlišuje mezi šesti různými typy zpráv MAC: požadavek na připojení, potvrzení připojení, poslaní nepotvrzených a potvrzených datových zpráv.

Tabulka 2.1: Přehled typů MAC zpráv [19]

Binární hodnota	LoRaWAN 1.0.4 Specification	Vysvětlení
000	Join request	OTAA žádost o připojení
001	Join accept	OTAA žádost o připojení
010	Unconfirmed data up	Uplink bez potvrzení
011	Unconfirmed data down	Downlink bez potvrzení
100	Confirmed data up	Uplink s potvrzením
101	Confirmed data down	Downlink s potvrzením
110	RFU	Vyhrazeno pro budoucí použití
111	Proprietary	Vlastní použití

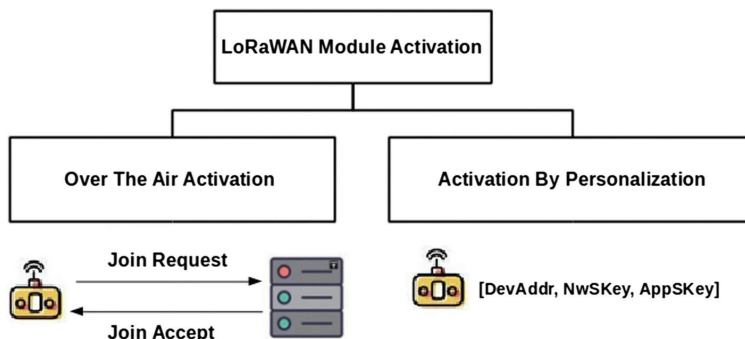
Zprávy, které jsou požadavkem na připojení a o přijetí se používají při aktivaci Over The Air Activation. Jsou popsány v následující kapitole.

Datové zprávy se používají k přenosu jak příkazů MAC, tak dat aplikací, které lze kombinovat společně do jedné zprávy. Zpráva s potvrzenými daty musí být potvrzena přijímačem, zatímco zpráva s nepotvrzenými daty nevyžaduje potvrzení.

Proprietární zprávy lze použít k implementaci nestandardních formátů zpráv, které nejsou kompatibilní se standardními zprávami, ale musí se používat pouze mezi zařízeními, která mají společné proprietární rozšíření.

2.1.3 Aktivace zařízení

Pro zahájení komunikace, musí být inicializováno koncové zařízení. LoRaWAN poskytuje dvě inicializační metody: Over The Air Activation (OTAA) nebo Activation By Personalization (ABP). Průběhy těchto aktivací jsou znázorněny na obrázku 2.7 a následně popsány.



Obrázek 2.7: Aktivace modulu LoRaWAN [20]

Over The Air Activation - OTAA

Tato metoda vyžaduje aktivaci na základě připojovací procedury, která v případě úspěšného navázání komunikace končí vygenerováním relačních informací. Při ztrátě relace (např. při resetu) je potřeba procedurou projít znova. V takovém případě je možné označit tuto metodu jako více náročnou na spotřebu. Pro tento proces, který probíhá mezi síťovým serverem a koncovým zařízením, musí být nejprve zařízení označeno klíči DevEUI, AppEUI a AppKey. Aktivace tedy probíhá ve dvou krocích – k výměně dvou zpráv se serverem a zařízením. Jedná se o zprávy join-request a join-accept. Následně jsou vygenerovány relační klíče NwkSKey a AppSKey.

Activation By Personalization - ABP

Specifikací méně doporučovaná metoda, která je ovšem hojně používána. Jak název napovídá, jsou kroky personifikace a aktivace sloučeny v jeden, to znamená, že nastavením, resp. personifikací zařízení, zároveň je provedena aktivace. Neprobíhá žádná komunikace se serverem a rovnou jsou odeslána aplikační data. Je nutné ručně nastavit DevAddr, NwkSKey a AppSKey. Díky tomu že, zařízení nemusí nijak navazovat komunikaci, ale rovnou může vysílat data, je při této metodě menší spotřeba energie.

Význam relačních klíčů je tímto statickým nastavením de facto redukován (neplní funkci relačně specifických a tedy proměnných klíčů). Zařízení inicializovaná metodou ABP jsou více náchylná na útok. Pokud jsou klíče vyzrazeny, veškerá budoucí komunikace už může být odposlouchávaná, a navíc může vést i ke kompromitaci zařízení ostatních (při nalezení vzorce pro generování těchto klíčů – hlavně u masového úniku klíčů).

Ačkoliv je ABP technicky méně bezpečná metoda, přesto je v praxi využívána. Důvodem je zpravidla horší přijímací schopnost zařízení než brány. To v praxi znamená, že odeslání uplink paketu je úspěšné, avšak downlink zprávy transportované z brány k zařízení nikoliv (pravděpodobnější je, že projde alespoň jeden paket, než že projdou oba). Tento fakt vytváří asymetrickou situaci a znemožňuje OTAA aktivaci, která tuto oboustrannou výměnu vyžaduje. [21]

Relační klíče

Po aktivaci jsou v koncovém zařízení uloženy následující informace: adresa zařízení (DevAddr), identifikátor aplikace (AppEUI), klíč síťové relace (NwkSKey) a klíč relace aplikace (AppSKey).

Vysvětlení využití jednotlivých klíčů:

- **End-device address (DevAddr):** Adresou koncového zařízení je 32bitový identifikátor zařízení v síti. Nejvyšších 7 bitů adresy slouží jako identifikátor sítě, zbylých 25 bitů slouží jako unikátní identifikátor koncového zařízení v síti.
- **Application identifier (AppEUI):** Je globální aplikační identifikátor, který jednoznačně identifikuje provozovatele aplikace, je tedy unikátní pro každou aplikaci. V každé aplikaci pak může být několik koncových zařízení. AppEUI je uložen v koncovém zařízení již před provedením aktivační procedury.
- **Network Session Key (NwkSkey):** Jedná se o klíč relace sítě specifický pro koncové zařízení. Používá jej síťový server i koncové zařízení k výpočtu a ověření MIC (message integrity code) všech datových zpráv, aby byla zajištěna integrita dat. Dále se používá k šifrování a dešifrování datových zpráv.
- **Application Session Key (AppSKey):** Tento klíč relace aplikace je také unikátní pro každé koncové zařízení, slouží k šifrování payloadu (užitečných dat). Síťový server nemusí AppSKey vůbec znát, protože k dešifrování payloadu dochází až v aplikačních serverech.
- **End-device identifier (DevEUI):** Každé zařízení disponuje již z výroby tímto unikátním identifikátorem. DevEUI vychází z adresního rozsahu IEEE EUI-64 a je ekvivalentem MAC adresy síťového zařízení v Ethernetu.
- **Application Key (AppKey):** Klíč unikátní pro každé koncové zařízení je přidělen provozovatelem aplikace a bývá odvozen z hlavního aplikačního klíče (root key), který je výhradně pod kontrolou provozovatele. AppKey je 128bitový klíč, který se využívá pro AES šifrování přenosu provozních klíčů a konfigurace v případě OTAA aktivace.

2.2 Příjem a zpracování dat

Možností, jak zprovoznit a připojit zařízení do sítě LoRa, je několik. Například komerční pokrytí signálem poskytují v ČR v největším měřítku České Radiokomunikace. Pro připojení zařízení nabízí několik balíčků, u kterých se cena liší podle počtu připojených zařízení. Další možností je využít projekt The Things Network (TTN). Jedná se o open-source komunitní projekt, kde připojení zařízení do sítě je zdarma, ale je omezeno množstvím vysílaných dat.

Dalším krokem toku dat je uložení přijatých dat do databáze. Jsou dvě možnosti. Buď přes HTTP, kdy je potřeba zajistit, aby server s databází byl dostupný na internetu (měl veřejnou adresu). Nebo pomocí MQTT. V tomto případě může TTN server sloužit i jako MQTT broker a pomocí MQTT klienta je možno data získat a uložit na jiný server.

2.2.1 Síťový server - The Things Network

The Things Network (TTN) je otevřená, decentralizovaná, komunitní platforma pro síť LoRaWAN, která vznikla kvůli vybudování cenově dostupných IoT řešení pro běžné uživatele.

Projekt The Things Network (TTN) je v Evropě docela rozšířen (Evropa – cca 9 000 bran, České republiky – 195) a poskytuje IoT připojení senzorům skrze LoRaWAN síť – komukoliv a bez poplatku. Stačí, když je v oblasti instalována nějaká brána, připojená (využívající) TTN síťový server. Je tedy možné buď připojit vlastní bránu nebo využívat brány ostatních uživatelů. TTN poskytuje připojení pro aplikace LoRaWAN a umožňuje registrovaným zařízením odesílat data, která jsou poté zpřístupněna na internetu. Brány jsou v rádiovém pokrytí LoRaWAN, ale je nutné je nakonfigurovat, aby bylo možné navázat spojení s TTN. [18], [22]

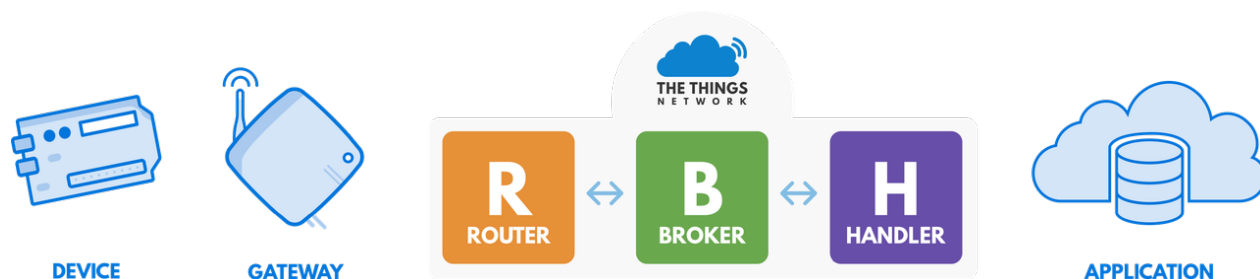
Platforma byla spuštěna 21.9.2015 v Amsterdamu, kde stačilo 10 bran pro pokrytí celého města. Účelem původně bylo upozornit vlastníky, že jim vniká voda do lodí v přístavu. Díky nízkým nákladům na vybudování a otevřenosti se síť velmi rychle celosvětově rozrostla a má široké využití.

Struktura TTN

Při návrhu struktury TTN se autoři zaměřili na škálovatelný a distribuovaný design, který umožňuje vysokou dostupnost, vysoký výkon a zabezpečení typu end-to-end a přitom plně vyhovuje specifikaci LoRaWAN. Na obrázku 2.8 jsou uvedeny klíčové komponenty sítě.

Router – služba, sloužící pro příjem zpráv z brány a předání dat Brokeru. Směrování se provádí na základě DevAddr. Dále provádí plánování vyslání downlink paketů. Vybírá, která brána bude vyžita k odeslání (počítá tzv. skóre, podle síly signálu, dostupnosti atp.).

Broker – Identifikuje zařízení, de-duplikuje provoz a předá paket Handleru, kde je aplikace registrována. De-duplikace paketů je potřeba z důvodu, že pakety mohou být přijaty od více bran, což je důležité (žádoucí). Ve výsledku data jsou sjednocena do jednoho paketu a metadata jsou rozšířena o informace ze všech zúčastněných bran.



Obrázek 2.8: The Things Network síťová architektura [22]

Handler – tato komponenta uživateli umožňuje zvolit způsob, kterým je paket dešifrován (na aplikační úrovni), poté je zpráva předána aplikaci. Jedná se o funkce psané v JavaScriptu, které lze použít k dekódování, převodu a ověření dat.

2.2.2 Omezení LoRaWAN

Síť LoRaWAN je primárně vytvořena pro nízkou spotřebu energie, velký dosah a poměrně nízkou nákupní cenu. Umožňuje odesílat jen malé množství dat a není vhodná pro přenos informací v reálném čase, zvuku, obrázků, ani videa. Nedoporučuje se přenášet JSON, nebo plain text. S narůstajícím časem vysílání roste spotřeba energie a obsazení sítě. Velikost odesílaných dat je omezena na 51 bajtů, ale záleží na nastavení parametrů (Spreading Factor, Bandwidth).

Síť musí splňovat kritéria pro vysílání uzákoněné daným státem. V rámci těchto pravidel, síť nemůže odesílat data příliš často. Maximální čas vysílání v jednom cyklu je stanoven na 1 %. To znamená, že pokud vysílání trvá dobu jednu sekundu, je možné ho znovu opakovat až po 100 s. Platforma The Things Network navíc stanovila maximální čas pro odesílání dat na pouze 30 sekund za 24 hodin a pro příjem na pouze 10 zpráv za 24 hodin na jedno zařízení. Tato omezení se vztahují pouze na veřejné komunitní síť.

2.2.3 Zpracování dat

Příchozí data nejsou na TTN serveru archivována. Je pouze možné si zobrazit doručené zprávy v době otevřeného náhledu dat. Po aktualizaci stránky je list vymazán a opět budou k dispozici pouze nově přijata data. Proto pro další zpracování je potřeba data uložit do databáze. Server TTN umožňuje několik integrací pro další zpracování. Jednou z nich je i MQTT, která je popsána v následujícím textu.

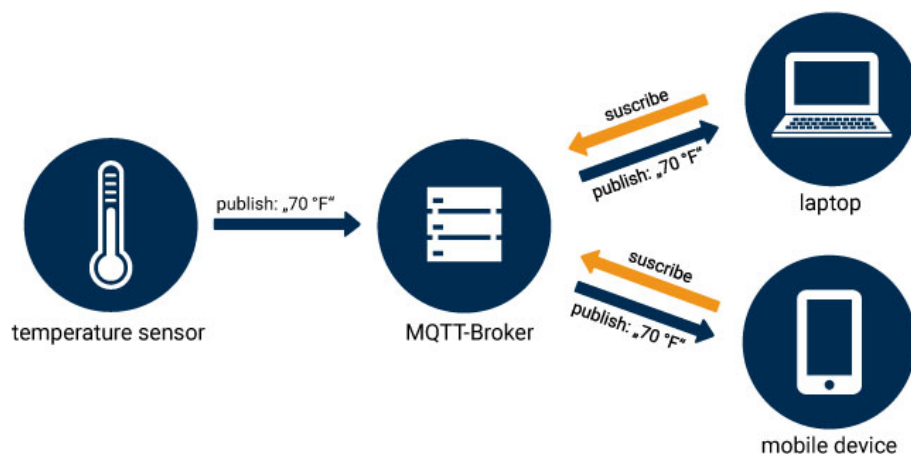
MQTT rozhraní

(dříve: Message Queuing Telemetry Transport, dnes MQ Telemetry Transport) je nenáročný protokol pro odesílání zpráv mezi zařízeními. Je vyvinut pro komunikace s nízkou zátěží na propustnost a výkon. MQTT je jednoduchý protokol pro předávání zpráv mezi klienty prostřednictvím

centrálního bodu – brokeru, který se stará o výměnu zpráv. Díky této nenáročnosti a jednoduchosti je možno ho jednoduše implementovat i do zařízení s „malými“ procesory. Byl navržen firmou IBM a poměrně rychle se rozšířil.

Obvykle funguje nad TCP / IP vrstvou. Je ideální pro IoT sítě a zařízení. Využívá model publish-subscribe, kde zařízení odesílající data se nazývají publisher a zařízení přijímající data se nazývají subscriber.

Odběratelé (subscribers) se přihlašují na jednotlivá témata u Brokera a zprávy jsou jim přeposílány. Zařízení mohou zastávat zároveň několik rolí pro odebírání a odesílání zpráv do více témat. Architektura MQTT je vysvětlena na následujícím obrázku (2.9).



Obrázek 2.9: MQTT architektura [23]

V našem případě bude jako Broker sloužit TTN server. Jako publisher (odesílatel) budou sloužit jednotlivá zařízení na stanovištích a subscribers (příjemce) bude v téhle práci nástroj Node-RED nainstalovaný na Raspberry Pi, který přichází data uloží do databáze nainstalovanou na stejném zařízení.

Node-RED – MQTT subscriber

Node-RED je vývojové prostředí pocházející od firmy IBM, nejčastěji se používající v oblasti IoT. Node-RED bylo už na počátku vydáno v roce 2013 jako opensource projekt. I kvůli tomu se utvořila velká komunita vývojářů sdílející své výsledky i s postupy a také diskutující své pokusy. Programování je zde založeno na flow modelu, kde se v grafickém prostředí spojují bloky předdefinovaného kódu (nody) pomocí vazeb určujících posloupnost jejich vykonávání (flow). Pokročilejší uživatelé mají navíc možnost vytvořit si vlastní nody s kódem v programovacím jazyce JavaScript.

Přístup k Node-RED je přes webový prohlížeč. K přenosu zpráv ve spojení s Node-RED je používán protokol MQTT, jenž je vhodný i pro komunikaci s TTN. Opět se zde využívá flow modelu k dosažení přijímání a odesílání požadovaných dat. Podpory se to v této oblasti dostalo například

i od firem jako Amazon či Microsoft. Software Node-RED je k dispozici primárně pro populární IoT Raspberry Pi a jejich OS Linux Raspbian, ale stejně dobře jej lze provozovat prakticky na jakémkoliv běžném počítači s OS Linux nebo případně i OS Windows.

Raspberry Pi

Toto zařízení zajistí funkci vzdáleného serveru. Raspberry Pi je malý jednodeskový počítač s deskou plošných spojů o velikosti zhruba platební karty. V roce 2012 byl vyvinut britskou nadací Raspberry Pi Foundation s cílem podpořit výuku informatiky ve školách a seznámit studenty s tím, jak mohou počítače řídit různá zařízení (např. mikrovlnná trouba, automatická pračka). Primárním operačním systémem je Raspbian. Cena se pohybuje v rozmezí 157–1 536 Kč (nejlevnější je Raspberry Pi Zero, nejdražší pak Raspberry Pi 4 Model B). Označení „Raspberry Pi“ je registrovanou ochrannou známkou.

MariaDB databáze

V rámci sběru a dalšího zpracování dat bude nutné data někde uchovávat. K tomuto účelu se bezesporu hodí určitá forma databáze. Pod pojmem databáze rozumíme určitý prostor či nástroj k uchování a organizaci dat. V praktické části je popsána práce s SQL relační databází na MariaDB serveru. Pod pojmem SQL pak rozumíme strukturovaný dotazovací jazyk, jenž v sobě implementuje problematiku výrokové matematiky. Používá se pro manipulaci s daty v databázích.

2.3 Vizualizace dat pomocí jazyka R

Pro zobrazení průběžných a finálních výsledku byl zvolen programovací jazyk R, pomocí kterého je možno vytvářet webové aplikace. R je programovací jazyk primárně sloužící pro statistické výpočty a vizualizaci dat, např. různé 2D/3D grafy nebo mapy.

Existuje více důvodů, proč byl zvolen zrovna jazyk R. Jedná se o jeden z nejpoužívanějších nástrojů v data science. R používají jak výzkumníci na univerzitách, tak i ve velkých komerčních firmách, jako třeba v Microsoftu, Facebooku, Googlu, Twitteru a dalších firmách. Díky tomu má R velkou komunitu, do které patří i spousta odborníků ve svém oboru, což nasvědčuje velkému množství kvalitně zpracovaných materiálů, ať už se jedná o knihy, články, návody na webu, video tutoriály nebo kurzy.

Dalším důsledkem existence velké a rostoucí komunity uživatelů je i to, že R obsahuje velké množství metod pro řešení téměř všech problémů v různých oblastech statistiky, ekonometrie a data science. Většina těchto funkcí je distribuována formou balíčků, které jsou uloženy na jednom z centrálních repozitářů.

Poslední, ale nikoli nejméně významnou výhodou, je to, že R je volně šiřitelný software, který funguje ve většině operačních systémů. Je zcela zdarma za podmínek GNU (General Public Licence).

R spadá do rodiny interpretovaných jazyků. To jsou jazyky, které pro svou činnost potřebují interpret. Interpret je software, který umožňuje provádění zdrojového kódu. Opakem jsou kompilované jazyky. Tedy jazyky, kde se zdrojový kód musí přeložit pomocí překladače do strojového kódu.

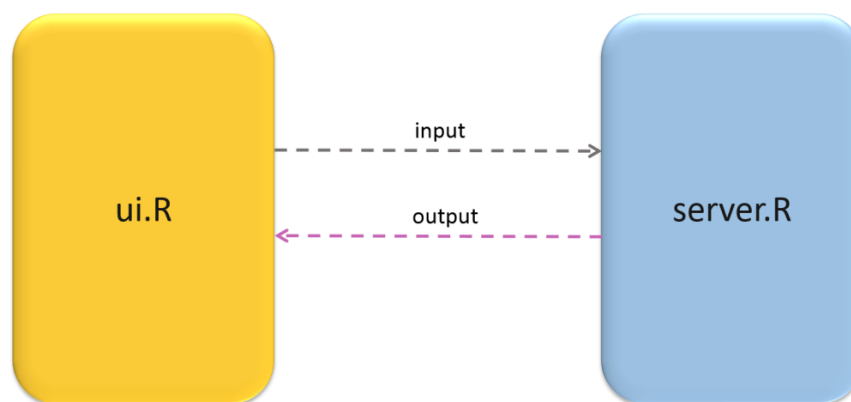
Důležitá součást jazyka R jsou knihovny. Pomáhají rozšiřovat možnosti jazyka a umožňují komunitě vývojářů přidávat funkcionalitu, která chybí v základním softwaru. Samotný jazyk R tvoří jen jádro základních funkcí. Knihovnam v jazyce R se říká packages (v literatuře taky označované jako framework, resp. rozhraní). Tyto knihovny je možné stáhnout z archivu CRAN („Comprehensive R Archive Network“) popřípadě i z jiných zdrojů jako github.com. Každá knihovna je obvykle zaměřena na určitou problematiku, jako třeba rozšíření množství grafů nebo knihovna pro rozsáhlejší manipulaci s daty a jiné. V následující části je uveden výpis knihoven použitých při vývoji aplikace.

2.3.1 Knihovna Shiny

Pro tvorbu webové aplikace byla použita knihovna Shiny. Tento balíček umožňuje tvorbu interaktivních webových aplikací v prostředí R. V Shiny je umožněna i podpora technologií jako HTML, CSS, nebo JavaScriptu, na základě čeho je možno aplikaci ještě více přizpůsobit potřebám uživatelů.

Pro tvorbu web Shiny aplikace pomocí knihovny Shiny je potřeba Shiny server, který umožňuje hosting Shiny aplikací.

Shiny aplikace je složena z dvou hlavních částí (obr. 2.10): uživatelské rozhraní (ui.R) a serverová část (server.R).



Obrázek 2.10: Schéma Shiny aplikace [24]

- ui.R (uživatelské rozhraní) – obsahuje kód, který se týká vzhledu aplikace a vstupních údajů od uživatele. Je možno vytvořit např. menu nebo přidávat textový/grafický obsah.
- server.R (serverová část) – funkční část aplikace (operace s daty, výpočty, zpracování), ke které koncoví uživatelé nemají přístup.

Od verze 0.10.2 je možné celou aplikaci uložit také do jednoho souboru s názvem app.R. Model rozdělení aplikace na dvě části je zde také přítomen, jediný rozdíl je v tom, že je vše uloženo do jednoho souboru.

V Shiny se používá reaktivního programování. Reaktivní programování je elegantní a výkonné paradigma programování. Klíčovou myšlenkou reaktivního programování je určit graf závislostí, aby se při změně vstupu automaticky aktualizovaly všechny související výstupy. Při změně vstupu není potřeba aktualizovat žádný výstup, protože Shiny to zajistí automaticky. Díky tomu je tok aplikace podstatně jednodušší. [25]

Je vhodné zdůraznit rozdíl mezi imperativním a deklarativním programováním. Tento rozdíl mezi příkazy a skripty je jedním z klíčových rozdílů mezi dvěma důležitými styly programování:

- Při imperativním programování je konkrétní příkaz proveden okamžitě. Toto je styl programování, při kterém je popsán výpočet pomocí posloupnosti příkazů, a tím je určen přesný postup.
- V deklarativním programování jsou vyjádřeny cíle na vyšší úrovni, nebo jsou popsána důležitá omezení. Program rozhodne o pořadí vykonávání funkcí. Toto je styl programování, který se používá v Shiny.

S imperativním kódem řeknete „Udělejte mi sendvič“. S deklarativním kódem řeknete „Zajistěte, aby v lednici byl sendvič, kdykoli se podívám dovnitř.“ Imperativní kód je asertivní; deklarativní kód je pasivně agresivní. [25]

2.3.2 Další důležité knihovny

Tidyverse

Tidyverse je kolekce základních R knihoven pro datovou analýzu. Knihovny obsažené v této kolekci jsou využívány při úpravě a interakci s daty. S daty je možné dělat celou řadu věcí, jako jsou podmnožiny, transformace, vizualizace atd. Tidyverse vytvořil Hadley Wickham a jeho tým s cílem poskytnout všechny tyto nástroje k čištění a práci s daty.

Jeden z hodně užitečných a populárních nástrojů je **Ggplot2**. Slouží pro vytváření grafů a vizualizací. Je možné vytvořit například krabicový graf, graf hustoty či graf časové řady.

Například díky knihovně **Dplyr** je možné manipulovat s daty. Jednou z největších výhod tohoto balíčku je, že je možno použít funkci kanálu „%>%“ ke kombinování různých funkcí v R. Od filtrování až po seskupování dat.

Dalším zajímavým balíčkem z téhle kolekce je **Purrr**. Vylepšuje sadu nástrojů R pro funkční programování (FP) tím, že poskytuje úplnou a konzistentní sadu nástrojů pro práci s funkcemi a vektory. Tato knihovna umožní nahradit mnoho složitých cyklů za kód, který se snadněji píše a je přehlednější.

Knihovna **Lubridate** je určena pro jednodušší práci s časem. Příkazy v R pro datum a čas jsou obecně neintuitivní a mění se v závislosti na typu použitého objektu data a času. Kromě toho metody, která jsou použita k úpravě času, musí být robustní vůči časovým pásmům, přestupným rokům, časům letního času a dalším zvláštnostem souvisejícím s časem, a R tyto funkce v některých situacích postrádá. Z toho důvodu vznikla knihovna Lubridate, která tyto požadavky zajistí.

Pomocí knihovny **Tidyr**, je snadnější organizování neboli „uklizení“ dat. Uklizená data jsou velice důležitá pro další zpracování. S uklizenými daty se snadno pracuje a je jednoduché je vizualizovat. Dvě nejdůležitější vlastnosti uklizených dat jsou:

- Každý sloupec je proměnná.
- Každé pozorování je v jednom řádku.

Shinydashboard

Existují i balíčky pro rozšíření jiných balíčků. Například balíček Shinydashboard rozšiřuje funkcionalitu balíčku Shiny. Tento balíček obsahuje soubor vlastností a funkcí pro vytvoření dashboardů v prostředí Shiny. Dashboardy jsou rozhraní, které se skládají z widgetů – jednotlivých vizualizací, které mají společný cíl. Často jsou jednotlivé objekty na sebe navázány a úprava (použití filtru) v jednom grafu dynamicky ovlivní i další grafy v reálném čase.

Leaflet

Balíček Leaflet, který je postaven na JavaScriptu, slouží pro vytvoření interaktivních webových map. Tyto mapy je možno použít přímo v Shiny aplikacích. Hlavní funkce tohoto balíku jsou: tvorba interaktivních map (posouvání, přibližování/oddalování mapy a pod.), zakreslování různých objektů do mapy (body, čáry nebo polygony).

Shinycssloaders

Shinycssloaders je malý balík, který obsahuje několik animací pro efekt načítání souborů, grafů, tabulek a dalších prvků.

Shiny semantic

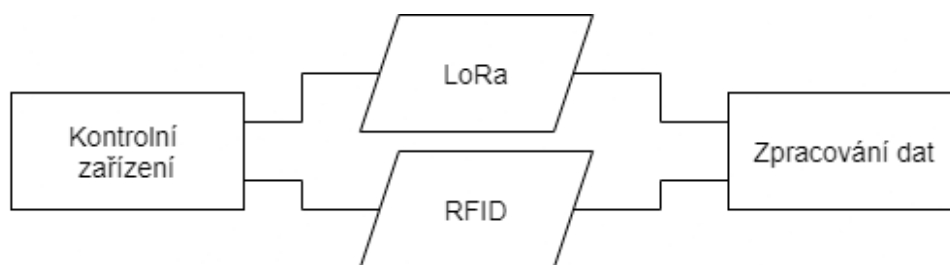
Shiny semantic rozšiřuje grafické možnosti uživatelského rozhraní. Obsahuje různé druhy šablon, se kterými se dá pracovat přímo v R, bez nutnosti použití html či css. Jsou to různé uživatelské vstupy, ikony, tabulky, drop-down menu a mnoho dalších.

Kapitola 3

Návrh a realizace infrastruktury pro sběr dat z kontrolních stanovišť s využitím bezdrátové technologie LoRa

Celý zpracovávaný systém je možné rozdělit na dvě části. Na obrázku 3.1 je uvedeno zjednodušené schéma celého systému. Skládá se z kontrolních zařízení a aplikace na zpracování dat. Přenos dat z kontrolních zařízení do aplikace je umožněn pomocí RFID čipů a LoRaWAN.

Tato kapitola pojednává hlavně o zpracování kontrolních zařízení umístěných v hracím poli. Na úvod této kapitoly jsou vyjmenovány požadavky, které má zařízení splňovat. Následuje výpis jednotlivých komponentů, ze kterých se skládá kontrolní zařízení. Následně je z použitých komponentů navrhována deska plošných spojů. Dále je popsán firmware, kde jsou podrobněji popsány nejdůležitější části kódu. Na závěr je rozebrána problematika pokrytí LoRa sítě v oblasti hracího pole.



Obrázek 3.1: Zjednodušené blokové schéma celého systému

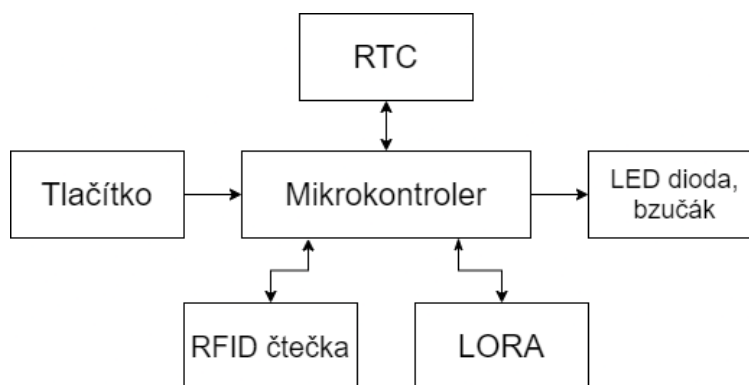
3.1 Požadavky na kontrolní zařízení

Pod pojmem kontrolních zařízení je myšleno zařízení umístěné na stanovištích. Jejich úkolem je, v případě, že účastník doběhne na stanoviště, zapsat tuto informaci na jeho RFID čip a vyslat tuto informaci přes LoRaWAN. Z toho vyplývají požadavky na tato zařízení:

- pomocí technologie LoRaWAN pošle data na server,
- pomocí RFID zapíše data na čip,
- má unikátní číslo,
- obsahuje modul reálného času, který zná aktuální čas,
- dá účastníkům dostatečnou zpětnou vazbu o provedení zápisu,
- změří teplotu uvnitř zařízení,
- změří napětí na bateriích,
- kryté ve vodotěsném obalu.

3.2 Použitý hardware v kontrolních zařízeních

Následuje výpis jednotlivých komponentů, které jsou použity pro konstrukci kontrolních zařízení. Mezi hlavní součástky patří mikrokontroler ATmega328P, RFID čtečka, LoRaWAN modul a modul reálného času.



Obrázek 3.2: Blokové schéma kontrolních zařízení

Mikrokontroler ATmega328P

Při výběru vhodného mikrokontroleru byl kladen důraz na dobrou dostupnost, nízkou cenu a širokou podporu ze strany výrobce a dostupného SW. Nejkritičtější parametry výběru byly spotřeba elektrické energie mikroprocesoru v režimu spánku a kompatibilita s napájecím napětím baterie. Všechny tyto požadavky splnil mikrokontroler ATmega328P a deska Arduino Pro Mini, která má pouze nezbytné součásti.

Mikrokontroler ATmega328P obsahuje 32KB paměti Flash, 1KB paměti EEPROM (Electrically Erasable Programmable Read-Only Memory) a 2KB paměti SRAM (Static Random Access Memory). Také obsahuje programovatelný sériový obvod USART, sériové rozhraní I2C, sériový port SPI, programovatelný watchdog timer s interním oscilátorem, a pět programovatelných režimů pro úsporu energie.

Tabulka 3.1: Základní parametry ATmega328P [26]

Mikrokontroler ATmega328P	
Napájecí napětí	2,7 až 5,5 V
Provozní proud	1,5 mA
Klidový proud	1 μ A
Provozní teplota	-40 až +125 °C

Vývojová deska Arduino Pro Mini 3V3 – Jedná se o jednu z nejmenších desek z dostupných druhů Arduina. Rozměry desky 34 mm x 18,5 mm. Deska obsahuje, obdobně jako obsažený mikrokontroler, 14 digitálních vstupů/výstupů a 8 analogových vstupů. Deska navíc obsahuje stabilizátor napětí. Oproti většině ostatních verzí Mini Pro neobsahuje integrovaný programátor, a proto pro naprogramování je potřeba použít některý USB – UART převodník. Maximální proudový odběr je uváděn 100 mA.

RN2483

Tento modul slouží pro komunikaci přes LoRaWAN. Výrobce je firma Microchip. Jedná se o Low-Power transceiver (TRANSMitter a RECEIVER) založený na technologii LoRa a využívající protokol LoRaWAN. Ovládá se přes rozhraní UART pomocí AT příkazů. Pracuje ve dvou frekvenčních pásmech 433 MHz a 868 MHz. Dosah sítě se udává až 15 km na předměstí a 5 km v zastavěné oblasti. Obsahuje integrovaný protokol LoRaWAN třídy A.

Tabulka 3.2: Základní parametry RN2483 [27]

Modul RN2483	
Napájecí napětí	2,1 až 3,6 V
Provozní vysílací proud	38,9 mA
Klidový proud	0,0016 mA
Provozní teplota	- 40 až +85 °C

RTC – Modul reálného času

Použitý modul obsahuje dva obvody DS3231 a AT24C32. První obvod DS3231 je označován jako RTC – Hodiny reálného času. A druhý obvod AT24C32 je paměťový modul typu EEPROM. Obvod RTC DS3231 vyniká oproti svým konkurentům vysokou přesností. Té je dosaženo integrovaným teplotně kompenzovaným krystalem, který má v rozsahu teplot 0-40 °C přesnost 2 ppm. Komunikace s celým modulem probíhá po sběrnici I2C. Tento modul také obsahuje senzor teploty.

Modul DS3231 uchovává informace o aktuálním čase a datu. Podporuje časové údaje jako: sekundy, minuty, hodiny, dny, měsíce i roky. Mezi užitečné funkce patří: korekce pro přestupný rok, datum ukončení, dva nastavitelné alarmy a kalendář, přepínání režimů AM a PM.

Tabulka 3.3: Základní parametry DS3231 [28]

Modul DS3231	
Napájecí napětí	3,3 nebo 5 V
Provozní proud	575 μ A
Klidový proud	110 μ A
Provozní teplota	0 až + 40 °C

RFID anténa a čip

Modul RFID čtečky – MF RC522 je čip pro bezkontaktní komunikaci na frekvenci 13.56MHz s možností čtení i zápisu. Umožňuje detekci chyb v přenosu ISO14443A, podporuje šifrování CRYPTO1. Obousměrná rychlost přenosu dat je až 424 kbit/s. Modul obsahuje SPI komunikační sběrnici a integrovanou anténu. Efektivní komunikační vzdálenost je až 3 cm.

Tabulka 3.4: Základní parametry MF RC522 [29]

Modul MF RC522	
Napájecí napětí	3,3 V
Provozní vysílací proud	13 až 26 mA
Klidový proud	10 - 13 mA
Provozní teplota	-20 až 80 °C

V této práci jsou použity pasivní RFID čipy. V takovém případě průběh komunikace mezi zařízením (čtečkou) a čipem je zahájen čtečkou, která vyšle impuls o délce asi 50 ms. Tento impuls je přijat anténou na čipu, která musí být naladěna na stejnou frekvenci jako anténa čtečky. Pomocí přijaté energie je nabit interní kondenzátor. Následně čip má energii pro vyslání dat zpátky ke čtečce. Délka přenášených dat je 128 bitů i se zabezpečovacím kódem a přenos těchto dat trvá přibližně 20 ms. Data jsou zachycena anténou čtečky a dekodována. Perioda mezi dvěma cykly je mezi 20 ms až 50 ms.

Na trh je uvedeno několik druhů RFID čipů. Z některých čipů je možno jen číst, další umožňuje jednorázový zápis a následné neomezené čtení a na třetí skupinu čipů je možno opakovaně zapisovat.

Pro tuto práci je použit poslední typ s možností opakovaného zápisu, konkrétně se jedná o čip MIFARE Classic 1K Memory Layout.

Vysvětlení jednotlivých pojmů: **MIFARE** je registrovaná ochranná známka NXP Semiconductors ze série čipů, které se běžně používají v bezkontaktních čipových kartách. **MIFARE Classic** používá vlastní protokol, který je v souladu s částmi 1–3 normy ISO / IEC 14443 typ A, s protokolem zabezpečení NXP pro autentizaci a šifrování. **1K Memory** – paměť o velikosti 1024 B.

1kB paměť čipu je organizována v 16 **sektorech** (od 0 do 15). Každý sektor je rozdělen na čtyři **bloky** (blok 0 až 3). Každý blok obsahuje 16 bajtů **dat** (od 0 do 15).

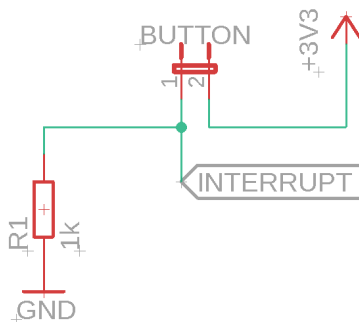
Ve výsledku máme tedy:

$$16 \text{ sektorů} \times 4 \text{ bloky} \times 16 \text{ bajtů} = 1\,024 \text{ bajtů} = 1 \text{ kB bajtů paměti}$$

I když je uvedena velikost paměti 1 024 B, není možné ji celou využít, z důvodu zabezpečení přístupu k datům. Třetí blok každého sektoru se nazývá Sector Trailer a obsahuje informace nazvané Access Bits, které umožňují přístup (číst a psát) ke zbývajícím blokům v sektoru. To znamená, že pro ukládání dat jsou k dispozici pouze tři dolní bloky (blok 0, 1 a 2), z toho plyne, že pro zápis je k dispozici pouze 48 bajtů z celkových 64. Také blok 0 sektoru 0 je známý jako Manufacturer Block obsahující údaje o výrobci (IC) a identifikátor (UID). [30]

Tlačítko

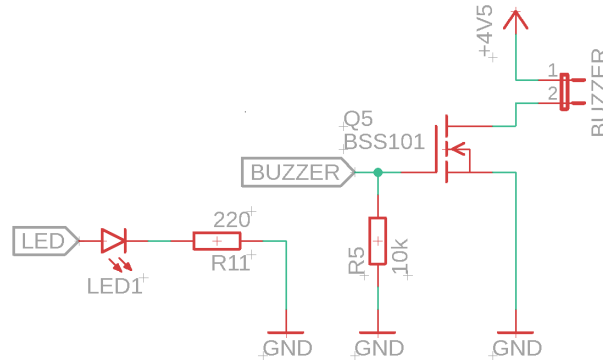
Hlavní funkcí tlačítka je přepnout zařízení z režimu spánku do aktivního režimu za pomoci vyvolání přerušení. Toto tlačítko bude sloužit hlavně pro uživatele – účastníky závodu, je tedy potřeba ho umístit tak, aby k němu byl umožněn dostatečný přístup i v případě, že je zařízení ukryto ve vodotěsném obalu. Zapojení je uvedeno na obrázku č. 3.3.



Obrázek 3.3: Zapojení – tlačítko

LED a piezo bzučák

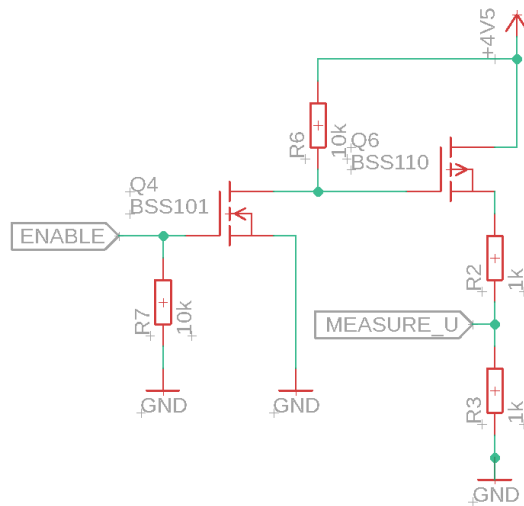
Zařízení obsahuje dvě LED a piezo bzučák. Tyto součástky budou sloužit jako ukazatel pro uživatele, že zařízení provádí svůj program. Jednou z nejdůležitější zobrazenou informací, je výsledek zápisu dat, zda zařízení provedlo zápis úspěšně či neúspěšně. Zapojení je na obrázku č. 3.4.



Obrázek 3.4: Zapojení – LED a bzučák

Napájení

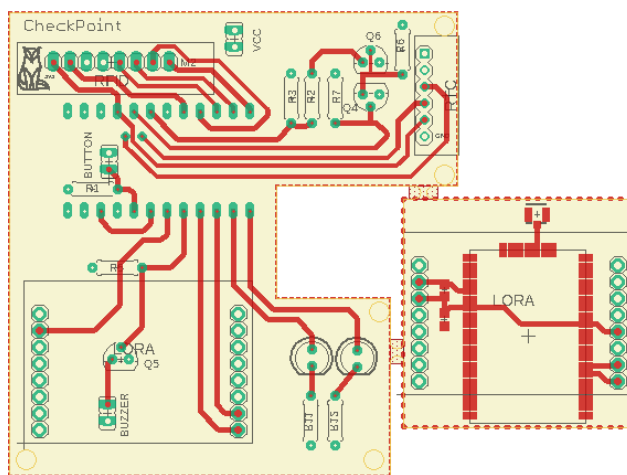
Zařízení je napájené z baterie. Vzhledem k tomu, že zařízení bude v provozu jen několikrát do roka, je nevýhodné pořizovat baterie s vyšší kapacitou, aby vydržely na více závodů, z důvodu samovolného vybíjení v době, kdy se zařízení nebudou používat. Proto jako zdroj energie byly zvoleny běžně dostupné jednorázové mikrotužkové AAA baterie. Napětí jedné takové baterie se udává 1,5 V. Vzhledem k tomu, že zařízení potřebuje minimální napětí 3,3 V je potřeba použít



Obrázek 3.5: Zapojení – měření napětí baterie

Jedním z požadavků na zařízení je měření napětí na bateriích. Vzhledem k tomu, že maximální hodnota napětí na bateriích je 4,5 V, ale mikroprocesor je 3,3 V bylo pro měření napětí potřeba použít dělič napětí. Při použití dvou stejných rezistorů bude naměřená hodnota poloviční. Po změření je nutné vynásobit hodnotu dvěma. Pro uzavření obvodu, jen v době potřebné k měření, byl použit tranzistor ovládající se pinem *ENABLE* z mikroprocesoru. Toto zapojení je uvedeno na obrázku č. 3.5.

Dalším krokem v realizaci zařízení je návrh zapojení a následně návrh rozložení desky plošných spojů. Tvorba všech schémat a DPS byla realizována pomocí programu Autodesk Eagle. Hlavní deska nemá jednoduchý obdélníkový tvar, neboť bylo potřeba vytvořit prostor pro tlačítko. V rozích jsou díry pro lepší uchycení desky v obalu. Také byla vytvořena malá deska pro modul LoRa. Obě vytvořené desky jsou na obrázku č. 3.6.



Na desce se nachází patice na připojení mikroprocesoru, RFID antény, RTC a LoRy. Pro všechny tyto moduly (kromě LoRy) byly již zakoupeny osazené desky s daným modulem. Pro jejich připojení tudíž stačí vyvést určité piny ve správném pořadí. Pro LoRu byla vytvořena další DPS, které je uvedena v pravé části obrázku. Tato deska obsahuje piny pro připojení k hlavní desce, plošky pro modul RN2483 a plošky pro konektor antény.

35

3.2.2 Finální podoba zařízení

Deska byla nejprve vyhotovena na půdě VŠB-TUO pro ověření funkčnosti a rozměrů a následně vyrobená ve firmě. Následovalo osazení desky. Na následujícím obrázku č. 3.7 je uvedena finální podoba zařízení. Hlavní deska je připevněna k víčku krabičky. Pro upevnění byly na 3D tiskárně vytisknuty hranoly, které drží desku ve stabilní poloze. Tlačítko je umístěno uprostřed víčka pro snadné ovládání. Hned pod ním se nachází RFID anténa. Dále byla v obalu vyvrtána díra pro vyvedení LoRa antény. V rámci této práce bylo vytvořeno pět funkčních zařízení.



Obrázek 3.7: Finální podoba zařízení

3.3 Popis Firmware

Funkce kontrolního zařízení je složena ze tří částí. Nejprve, když přijde účastník na stanoviště a stiskne tlačítko, je pomocí přerušení aktivováno zařízení. Dále RFID anténa po přiložení čipu provede zápis na čip a na závěr jsou vyslány data přes LoRaWAN na server. Jednotlivé úkony jsou rozebrány v následujícím textu.

3.3.1 Ovládání a signalizace zařízení

Jak už bylo zmíněno, zařízení obsahuje tlačítko, které při stisknutí způsobí přerušení a aktivuje zařízení. Po jeho stisknutí je možno přiložit k zařízení čip. Následně se provede zápis o průchodu stanovištěm. V případě, že vše proběhne správně, je spuštěna jiná signalizace, než když nastane chyba při zápisu. V době, kdy na stanovišti nikdo nebude, bude zařízení přepnuto do režimu spánku.

Je velmi důležité, aby zařízení v dostatečné míře oznamovalo, co se aktuálně děje. V případě této práce zařízení nemá žádný displej, takže signalizace bude probíhat jen za použití LEDek a piezo bzučáku. Hned po aktivaci zařízení je rozsvícena jedna z LEDek – signál, že zařízení je aktivováno. Následně po zahájení komunikace RFID antény dioda začne blikat. Při zahájení komunikace s čipem LEDka přestane blikat a je sepnuta až do konce komunikace. Signalizace je ukončena signálem podle výsledku zápisu. Buď zápis proběhl v pořádku – rozsvítí se druhá LEDka a ozve se bzučák po dobu půl sekundy, v případě chyby dioda několikrát zabliká a ozve se opakovaný krátký signál z bzučáku.

Watchdog timer

Vzhledem k občasnému zaseknutí zařízení byl zaimplementován watchdog. Časovač watchdog (WDT) je hardwarový časovač, který automaticky generuje reset systému, pokud jej hlavní program opomene pravidelně obsluhovat. Často je využíván k automatickému resetování zařízení, které přestane reagovat kvůli chybě softwaru nebo hardwaru, kdy tento reset nemůže provést obsluha zařízení.

Pokud je tento časovač povolen, je nutné ho v běhu programu po určité době resetovat (příkazem v programu). Jestliže se program zasekne, nebude tudíž příkazem v programu resetován, dojde k aktivaci resetovací sekvence CPU. Tato funkce zajistí resetování zařízení, a tím dojde k opětovnému bezchybnému spuštění obvodu.

V případě navrhovaného zařízení jeho činnost (obsluha jednoho závodníka) trvá přibližně 10 s. U mikrokontroleru ATmega328P je možnost nastavit wtg v rozmezí od 15 ms do 8 s. Nejjednodušší je zvolit 8 sekund a následně při aktivní činnosti alespoň dvakrát tento časovač resetovat.

Režim spánku a přerušení

Z důvodu, že zařízení je napájeno z baterie a většinu svého provozního času nebude vykonávat žádnou činnost, je potřeba pracovat s přepínáním režimu činnosti. Následuje postup přepnutí

do režimu nižší spotřeby u mikroprocesoru ATmega328P. Postup pro další komponenty je uveden v následujících odstavcích.

Když se u mikroprocesoru přepne režim do režimu spánku, tak se vypnou všechny nepotřebné komponenty, čímž se sníží jeho spotřeba. Mikroprocesor ATmega328P má několik režimů spánku. Čím hlubší spánek je zvolen (více vypnutých periférií), tím omezenější jsou možnosti, jak čip vzbudit. Ten nejvíce účinný režim, to znamená, že po jeho přepnutí má modul nejmenší spotřebu, je režim `SLEEP_MODE_PWR_DOWN`. V tomto režimu je jediný způsob, jak je možno mikroprocesor aktivovat je pomocí přerušení. Je ale potřeba, ještě před uspaním mikroprocesoru povolit přerušení na pin, ke kterému je připojené tlačítko.

3.3.2 RFID komunikace

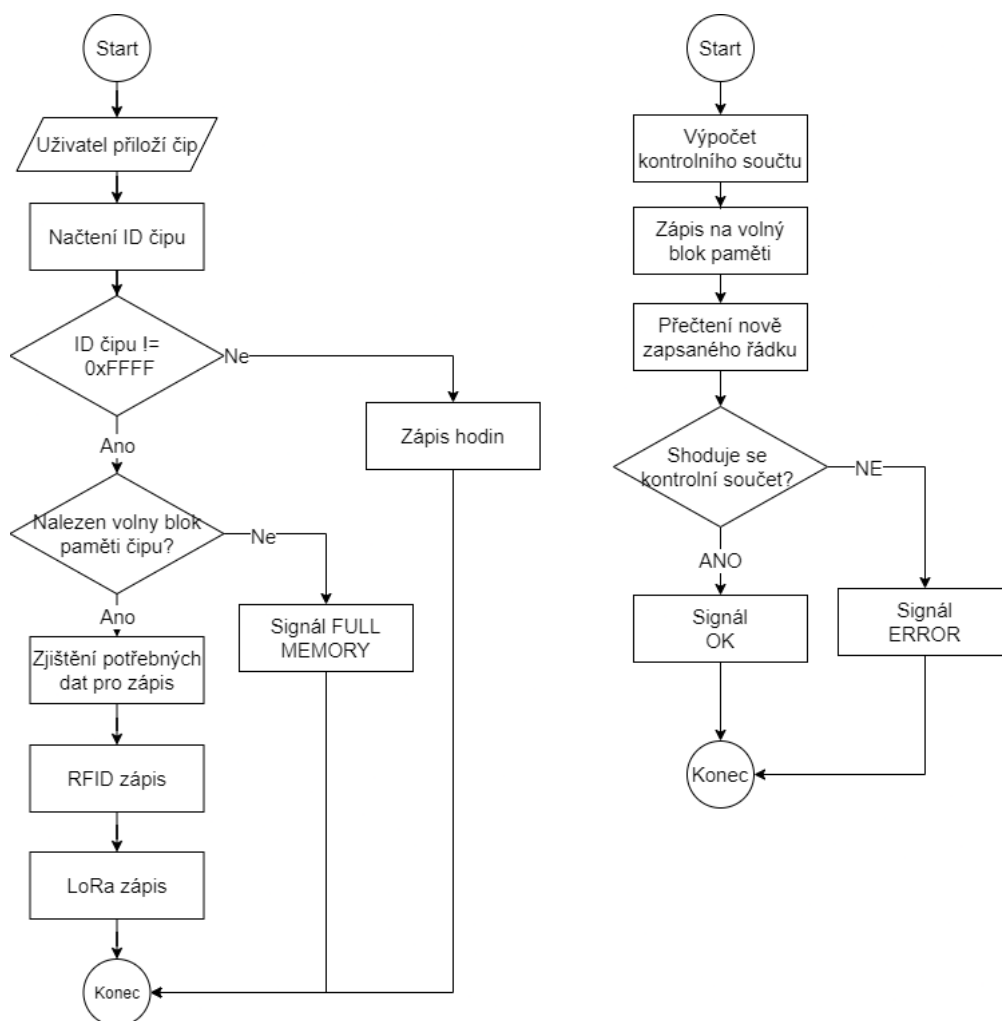
Modul MFRC522 je k mikroprocesoru připojen pomocí SPI sběrnice. Na začátku běhu programu se nachází inicializace komunikace přes protokol SPI a komunikace s RFID čtečkou. Následně v hlavním programu se nejprve zkontroluje výskyt RFID čipu v okolí antény. V případě úspěšné detekce čipu se zkontroluje jeho správné načtení. Dále může proběhnout obsluha čipu, která je znázorněná v diagramu v následujícím obrázku č. 3.8.

Po přiložení čipu k RFID čtečce se jako první zjistí ID čipu, které se nachází na prvním bloku paměti čipu. V případě, že hodnota je `0xFFFF`, jedná se o servisní čip, který má v následujícím bloku paměti uložen „aktuální čas“. V takovém případě se čas z paměti vyčte a aktualizuje se čas v zařízení. Více informací i této problematice je uvedeno v kapitole 3.3.4.

V opačném případě, tedy pokud je ID rovno jakékoliv jiné hodnotě, se postupně procházejí jednotlivé bloky paměti čipu (od 4. po 64.), až se nalezne prázdný blok bez zápisu. V případě nenalezení prázdného bloku, což znamená, že paměť čipu je již plná, program skončí a zazní zvukový signál. Tohle by se při reálném použití nemělo stát, ale při vývoji to byl častý problém. V případě nalezení volného bloku dat už nic nebrání provést zápis.

Před samotným zápisem je ještě potřeba zjistit údaje, které se budou zapisovat na čip, případně vysílat pomocí LoRaWAN. Pro zápis na RFID čip je potřeba znát ID volného bloku paměti, ID daného zařízení a aktuální čas. Pro vyslání dat pomocí LoRaWAN je navíc ještě potřeba znát ID přiloženého čipu, napětí naměřené na bateriích a naměřenou teplotu. Více o LoRaWAN komunikace je popsáno v další části.

Vlastní zápis je uveden na obrázku 3.8 vpravo. Nejdříve se provede součet všech zapisovaných hodnot. Tento součet bude následně sloužit jako kontrola správného zápisu dat. Následně se data zapíše na volný blok paměti i s kontrolním součtem. Následuje kontrola správného zápisu. Hodnoty se nyní z daného bloku dat vyčtou a jejich součet se porovná s kontrolním součtem. V případě shody se provede signál *OK*, případně, že jsou hodnoty různé – což znamená, že někde při zápisu či čtení nastala chyba, se provede signál *ERROR* a data je potřeba zapsat znovu.



Obrázek 3.8: Diagram RFID komunikace – vlevo celý průběh, vpravo jen zápis dat na čip

Po dokončení komunikace pomocí RFID antény je vhodné ukončit komunikaci a její zabezpečení. I po tomto kroku je proudový odběr stále vyšší. Pro snížení je potřeba modul uspat. Jednou z možností je softwarové řešení, neboli „soft power down“, při kterém je potřeba nastavit vhodnou hodnotu v registrech.

Datový formát

Množství dat, které se zapisuje na čip, není velké. V podstatě by stačilo zapisovat na čip pouze ID stanoviště, na kterých účastník byl. Nicméně, pro správné vyhodnocení závodu, je třeba znát dobu, kterou účastník stráví na úkolovém stanovišti. Proto je důležitý zápis času příchodu a odchodu účastníka na stanoviště. Z tohoto důvodu se na čip zapisuje informace o aktuálním čase do čtyř bajtů. Následující dva bajty jsou rezervní, pro případ zápisu dalších dat a také z důvodu, aby celý zápis měl osm bajtů. Na posledním bajtu je uveden kontrolní součet. Což znamená, že v případě

potřeby se na jeden blok paměti, který má šestnáct bajtů, vejdu až dva zápisy. Datový formát je uveden na obrázku 3.9.

BYTE	0	1	2	3	4	5	6	7
DATA	ID CP	den	hodiny	minuty	sekundy	rezerva		checksum

Obrázek 3.9: Datový formát RFID

0. bajt – ID daného stanoviště

1. až 4 bajt – Aktuální čas v pořadí den, hodiny, minuty a sekundy

7. bajt – Kontrolní součet

3.3.3 LoRaWAN komunikace

Modul RN2483 je připojen k mikroprocesoru pomocí UART sběrnice. Vzhledem k tomu, že při komunikaci mezi zařízením a počítačem, například při nahrávání nové verze firmware, je také používán UART, je potřeba komunikaci pro modul RN2483 oddělit. Pro tyto účely je použita knihovna SoftwareSerial, pomocí které je možné mít více sériových komunikací. Tato knihovna nám umožní použít kterýkoliv pin na mikroprocesoru jako pin TX a RX.

V případě této práce je LoRa modul autentizován do sítě pomocí metody ABP. Aby bylo možné modul připojit do sítě, je potřeba IoT serverem vygenerovat parametry, následně je napsat přímo do kódu pro zařízení a po spuštění je vyslat do modulu. Následně je možno se připojit do sítě a posílat zprávy.

Tabulka 3.5: Základní AT příkazy pro komunikaci s LoRa modulem [31]

AT příkaz	Vysvětlení
0x00 0x55	Změna UART rychlosti na 9 600 bps ukončení režimu spánku
mac set devaddr <address> mac set nwkskey <address> mac set appskey <address>	Nastavení parametrů pro připojení
mac join abp	Připojení do sítě metodou abp
mac tx uncnf 1 <data>	Odeslání dat
sys sleep <time>	Režim spánku [ms] hodnoty od 100 do 4 294 967 296

V případě nečinnosti je možné modul přepnout do režimu spánku. Při nastavování režimu spánku je potřeba i uvést čas, na jak dlouho má být modul uspán. Maximální nastavitelná hodnota je přibližně $4 \cdot 10^9$ ms, to odpovídá asi 50 dnům, což pro naše potřeby naprosto stačí. Při nastavení

takové vysoké hodnoty se nemůže stát, že by se modul probudil sám. Následně je možné ho probudit změnou UART rychlosti. Přehled základních AT příkazů je uveden v tabulce č. 3.5.

Datový formát

V kontrolních zařízeních je modul LoRa použit jen pro vyslání dat. Vysílaná data obsahují 10 bajtů. Datový formát je uveden na obrázku 3.10 a následuje podrobnější popis jednotlivých bajtů.

BYTE	0	1	2	3	4	5	6	7	8	9
DATA	0x11	ID CP	ID čip		hodiny	minuty	sekundy	baterie	teplota	

Obrázek 3.10: Datový formát LoRa

- 0. bajt** – Nultý bajt je vždy nastaven na hodnotu 0x11 – ať server pozná, že se jedná o zápis průchodu stanovištěm.
- 1. bajt** – V dalším bajtu je uvedeno ID stanoviště.
- 2. a 3. bajt** – Na dalších dvou bajtech je uložena identifikační hodnota čipu účastníka, který dané stanoviště navštívil.
- 4. až 6. bajt** – Na těchto třech bajtech je postupně uložena hodnota aktuálního času. Jedná se o hodnotu hodin, minut a sekund.
- 7. bajt** – Na sedmý bajt se ukládá hodnota napětí baterie. Protože naměřená hodnota obsahuje jedno desetinné číslo a je někde v rozmezí 3,3 – 4,5 je posílaná hodnota vynásobená 10, tudíž bude vysláno jen celé číslo.
- 8. a 9. bajt** – Na posledních dvou bajtech je uložena hodnota teploty. Opět se jedná o desetinné číslo, proto uvedená hodnota je vynásobená deseti.

3.3.4 Další pomocné funkce

Pro zajištění správného fungování zařízení (hlavně RFID čipů) jsou potřeba i další funkce. Jejich úkolem je vyčtení zapsaných dat a uložení do databáze pro další zpracování a připravení čipů na nový závod – vymazání paměti. Pro tyto účely je vytvořené další zařízení, které obsahuje RFID čtečku a konektor USB na připojení k počítači. Všechny tyto moduly jsou ovládány pomocí webové aplikace, která je popsána v kapitole 5.3.4

Načtení dat z čipů

Důležitou částí pro vyhodnocení závodu je i načtení dat o průchodech přes stanoviště, která mají účastníci uložena na RFID čipech. Proto po doběhnutí účastníků do cíle je potřeba tyto čipy vysbírat a následně z nich data vyčíst. Vyčítání dat se provede po přiložení čipu k zařízení, které tato

data pošle po sériovém portu (USB). Při vyčtení každého bloku dat se provede kontrola pomocí kontrolního součtu. V případě chyby se řádek přečte znova. Po opětovném špatném přečtení se pokračuje čtením dalšího řádku.

Mazání dat z čipů

Pro použití čipů na dalších závodech je potřeba vymazat jejich paměť. Pro tyto potřeby je vytvořen jednoduchý kód, který projde jednotlivé bloky paměti čipu a zapíše do nich nulovou hodnotu. Následně projde znova všechny řádky a zkontroluje, že zápis byl proveden správně. Na závěr uživatele informuje a výsledku pomocí bzučáku a LED.

Zápis aktuálního času do zařízení

Jedna z otázek je, jak nastavit aktuální čas ve všech zařízeních, kterých je asi 30 kusů. Modul hodin reálného času má záložní zdroj energie v podobě vlastní baterie. Ta může dodávat modulu energii, i když je hlavní napájení odpojeno. Takže poskytované informace modulem tedy mohou být vždy aktuální i při odpojení hlavní baterie. Ale vzhledem k tomu, že zařízení se budou používat jednou nebo dvakrát do roka, je opotřebení baterie v čase, kdy se zařízení nebude používat, nevýhodné. V tom případě pokaždé při odpojení baterie se čas vynuluje.

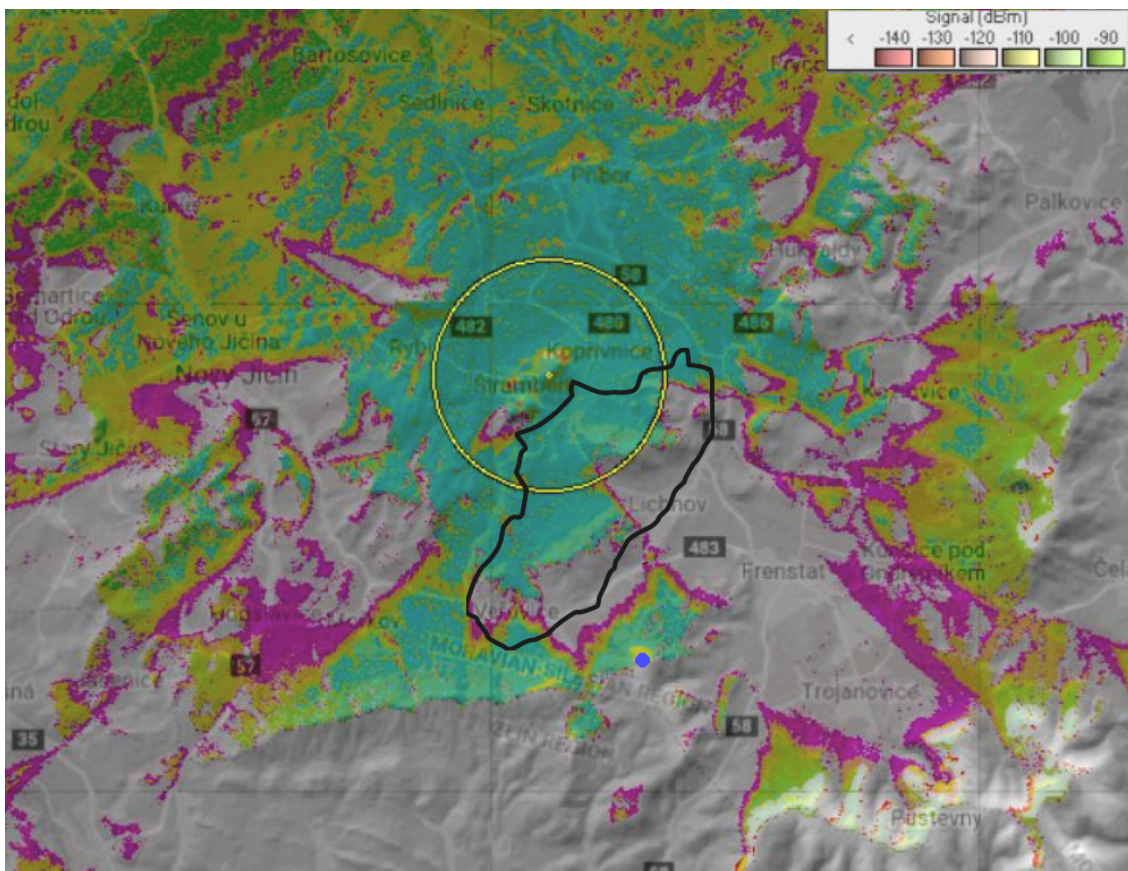
Jedna z možností, jak před každým závodem zapsat aktuální čas do zařízení, je přenést data pomocí RFID čipu. Zařízení, které je připojeno k počítači, zapíše aktuální čas na čip a následně tento čip přiložíme k zařízení, do kterého chceme aktuální čas zapsat. Pro více zařízení je vhodné, provést opakovaně zápis aktuálního času na čip. Tento postup je nutný z důvodu zachování alespoň částečné přesnosti času v zařízeních. To znamená, že po zapsání hodin na čip, je možno tento čas zapsat do maximálně pěti dalších zařízení. Pokud chceme zapsat čas i do dalších zařízení, je nezbytné na čip znovu zapsat aktuální čas z počítače. Pro zápis času z čipu na zařízení je potřeba, aby ID čipu bylo rovno 0xFFFF (nejvyšší možná hodnota). V takovém případě zařízení očekává, že se na druhém řádku nachází hodnota času, kterou nastaví jako aktuální čas v modulu RTC. Tenhle postup nejspíše nezaručí velkou přesnost, ale to pro potřeby našeho systému je dostačující.

Zápis ID čipu

Pro jasnou identifikaci účastníků je nutné, aby každý čip měl své ID. Nejedná se o často používanou funkci, ale i tak je potřeba. K tomu je určen modul, který se na začátku vždy zeptá na hodnotu, kterou má zapsat, a následně po přiložení čipu tuto hodnotu zapíše na první blok paměti do dvou bajtů. Je tedy možné volit označení od 0 – 65 535. Toto číslo je uvedeno i přímo na obalu čipu.

3.4 Pokrytí hracího pole sítí LoRaWAN

V době vývoje systému nebylo herní území pokryto signálem. V rámci projektu, do kterého je zapojená VŠB-TUO, bude instalována LoRaWAN brána na rozhledně poblíž daného území. Konkrétně se jedná o rozhlednu na Bílé hoře (557 m n. m., Štramberk). Tato brána bude připojena do sítě TTN. Vzhledem k tomu, že hrací pole se nachází docela v kopcovitém terénu, signál z této antény pokrývá asi 60 % hracího pole. Největší překážka je kopec nazývaný Červený kámen. Na obrázku 3.11 je uvedena mapa pokrytí z antény s vyznačením hracího pole. Dalším krokem pro pokrytí zbylé části hracího pole, by mohla být instalace další brány na blízkém kopci Velký Javorník (918 m n. m., Bordovice). Možné umístění brány na tomto kopci je vyznačeno na obrázku č. 3.11 modrým bodem. Mapa pokrytí byla vykreslena pomocí SW Radio Mobile.



Obrázek 3.11: Mapa pokrytí oblasti s herním polem

Kapitola 4

Analýza energetické náročnosti jednotlivých komponent infrastruktury

Během práce byla provedena analýza energetické náročnosti jednotlivých komponentů infrastruktury. Je poměrně náročné podle návrhu odhadnout konečnou spotřebu energie, protože často se reálná spotřeba modulů liší od spotřeby udávané výrobcem. Analýza byla provedena v několika krocích. Nejprve byl změřen odebíraný proud jednotlivých modulů v zařízení v aktivním režimu či režimu spánku. Dále proběhlo měření proudu pro kompletní zařízení a následně byl stanoven odhad výdrže na baterii.

Pro napájení zařízení byly zvoleny tři mikrotužkové AAA baterie s celkovým napětím 4,5 V. Při měření bylo zařízení napájeno přes bateriový konektor totožným napětím 4,5 V. Zařízení obsahuje stabilizátor, pomocí kterého jsou moduly napájeny stabilizovaným napětím 3,3 V.

Důležitým faktorem při otázce snížení spotřeby energie je, jak často bude zařízení vykonávat nějakou činnost. Jinak řečeno, jak často nějaký účastník doběhne na stanoviště a bude potřebovat zapsat záznam na čip. Z poznatků z předchozích ročníků můžeme určit, že:

- akce se koná jednou ročně,
- závod probíhá v sobotu od 8:00 do 16:00,
- většina zařízení je umístěna už v pátek večer,
- doba jednoho zápisu trvá přibližně 10 sekund,
- závodu se účastní okolo 100 závodníků.

Z výše uvedených informací lze odvodit, že pohotovostní doba zařízení je okolo 24 hodin, zatímco doba, po kterou bude provádět nějaké instrukce, podle vzorce 4.1 je $10 \cdot 100 = 1000$ s, což je přibližně 17 minut.

$$Provozní_doba = počet_účastníků \cdot doba_zápisu \quad (4.1)$$

4.1 Měření odběru jednotlivých komponent

Celková spotřeba závisí na spotřebě jednotlivých komponentů v daných režimech. Pro co nejnižší spotřebu je provedeno měření i jednotlivých komponentů. Naměřený proudový odběr je uveden v tabulce č. 4.1. Hodnoty jsou změřeny při vstupním napětí $U = 4,5 \text{ V}$.

Zařízení pracuje zejména ve dvou režimech – vysílání a spánek. V režimu spánku zařízení neprovádí žádné akce, pouze čeká na vyvolání přerušení, po kterém se přepne do aktivního režimu – vysílání. V režimu vysílání zařízení provádí hlavně komunikaci s RFID čipem a vyslání dat pomocí LoRy.

Tabulka 4.1: Proudový odběr jednotlivých modulů

Modul	Vysílání [mA]	Spánek [mA]
Mikroprocesor	10,16	1,62
RTC	0,43	0,43
RFID	22,6	0,00
LORA	5,56	0,00
Součet	38,75	2,05

Pro $U = 4,5 \text{ V}$

Při měření byly ze všech modulů, kromě desky u mikroprocesoru, odebrány LED. Z toho důvodu je u mikroprocesoru vyšší proudový odběr i v případě režimu spánku. RTC nemá žádný režim spánku. Je ale možnost, pro snížení spotřeby, napájet modul přes V_{BAT} , díky čemuž klesne spotřeba na $1 \mu\text{A}$, ale není pak možné použít některé jeho funkce. V případě LORY se jedná o průměrnou hodnotu, neboť, jak bude vidět níže (při popisu průběhu), má průběh vysílání tři velké impulzy. U položky součet se jedná pouze o teoretickou hodnotu, neboť například RFID a LORA nevysílá ve stejný okamžik najednou.

4.2 Měření odběru celého zařízení

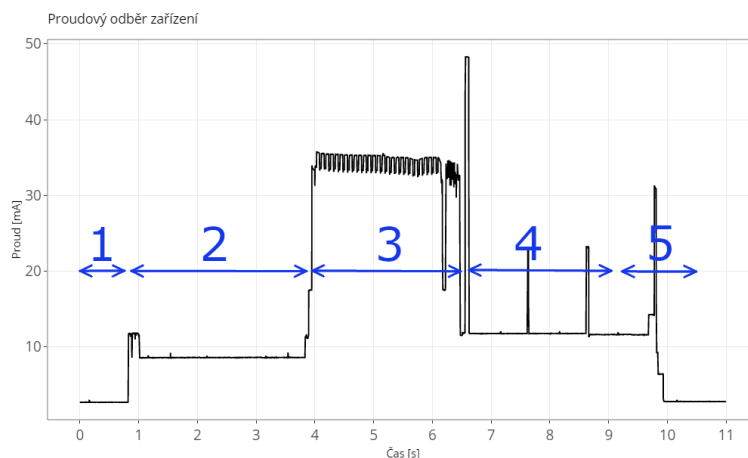
V této podkapitole je zdokumentováno měření odběru celého zařízení. Naměřené hodnoty jsou tedy celkové včetně možných ztrát v zařízení. Jako první je uveden popis odběru obslužení jednoho závodníka. Následuje simulace reálného provozu – proudové odběry pro různé množství účastníků, kteří navštíví stanoviště se zařízením za určitý čas. Z těchto měření byla odvozena výdrž baterie pro dané scénáře.

Měření spotřeby bylo provedeno analyzátozem výkonu N6705C, od firmy KEYSIGHT. Analyzátor umožňuje velmi přesně změřit odběr zařízení při zvoleném napětí. Vzorkovací perioda T byla

nastavena na interval 1 ms, což odpovídá frekvenci měření 1 000 Hz, tedy 1 000 vzorků za sekundu. Ke zpracování naměřených dat byl využit jazyk R, pomocí kterého byly vykresleny průběhy odběru proudu.

4.2.1 Popis průběhu měřeného odběru

Následuje popis průběhu odběru proudu při obsluze jednoho závodníka. Průběh lze rozdělit do pěti částí, během kterých jsou vykonávány určité části programu. Samotný průběh odběru proudu je vykreslen na obrázku č. 4.1.



Obrázek 4.1: Proudový odběr obsluhy jednoho závodníka

V první části (označena jedničkou) je zařízení v režimu spánku. Během této části má zařízení průměrný odběr 2,66 mA. Stejný průběh je možné vidět na konci páté části.

Na začátku druhé části bylo stisknuto tlačítko, což vyvolá přerušení, které aktivuje zařízení. Ihned po aktivaci zařízení je rozsvícena notifikační dioda. Dále zařízení provede inicializaci modulů. Průměrný odběr této části je 8,82 mA.

V třetí části je možné vidět dva důležité jevy. V průběhu je znát velký nárůst odběru a dále malé skokové změny hodnot. Velký nárůst je způsoben vysláním RFID antény. RFID anténa pro komunikaci s čipem musí vyslat větší množství energie, kterou se pak nabije přiložený čip. V čase 6 s je možné vidět pokles hodnoty. Anténa navázala komunikaci s čipem a nyní se zjišťují potřebné údaje pro zápis. Opětovný nárůst hodnot značí provedení zápisu dat na čip. V případě, že se do určitého časového limitu k RFID anténě nepřiloží žádný čip, zařízení se poté přepne do režimu spánku. Skokové změny hodnot jsou způsobeny signalizační diodou, která bliká v případě, kdy zařízení čeká na přiložení čipu. Průměrný odběr této části je 33,89 mA.

V další čtvrté části je vidět průběh LoRa komunikace. Výrazný impuls na začátku představuje odeslání dat. Protože zařízení pracuje ve třídě A, po odeslání dat jsou otevřeny vždy dvě naslouchací okna, určené pro příjem dat. První časový interval, pro otevření prvního naslouchacího okna, je

nastaven na hodnotu 1 000 ms a dále, po uplynutí dalších 1 000 ms se spustí druhé naslouchací okno. Průměrný odběr této části je 13,15 mA s nejvyšší hodnotou 48,31 mA.

V poslední páté části se uživateli signalizuje správnost provedení zápisu. Poslední výrazný odběr je způsoben sepnutím obvodu s bzučákem. Následuje uspání celého zařízení. Průměrný odběr této části je 9,32 mA s nejvyšší hodnotou 31,31 mA.

4.2.2 Simulace reálného provozu

V následující části je naměřen odběr pro různé množství návštěvníků, kteří navštíví stanoviště za určitý čas. Čas je v tomto případě vždy 10 minut. Z naměřených výsledků byla stanovena průměrná spotřeba kontrolního zařízení. Graf naměřených hodnot je v příloze č. A.3. Po celou dobu bylo nastaveno konstantní napětí $U = 4,5$ V.

Tabulka 4.2: Porovnání různých scénářů použití zařízení

		0	1	3	3 _h	5	5 _h	10
I_avg	[mA]	2,66	2,92	3,30	3,34	3,63	3,68	4,38
I_max	[mA]	4,01	48,31	48,14	48,02	47,61	47,77	48,09
I_min	[mA]	2,62	2,55	2,72	2,72	2,69	2,72	2,72
T_vysílání	[s]	0,00	9,11	25,19	25,82	41,04	41,38	79,87
T_spánek	[s]	600,00	590,89	574,81	574,18	558,96	558,62	520,13
I_avg_vysílání	[mA]	0,00	17,26	15,77	16,21	15,45	15,04	14,93
I_avg_spánek	[mA]	2,66	2,70	2,76	2,76	2,76	2,76	2,76
P_avg_vysílání	[mW]	0,00	76,86	71,10	73,12	69,12	67,68	66,91
P_avg_spánek	[mW]	11,97	12,15	12,42	12,42	12,42	12,42	12,42
E_vysílání	[mWs]	0,00	700,19	1791,01	1888,09	2836,68	2800,59	5344,50
E_spánek	[mWs]	7182,00	7179,31	7139,14	7131,32	6942,28	6938,06	6460,02
E_celková	[mWs]	7182,00	7879,51	8930,15	9019,40	9778,97	9738,66	11804,52
Pro $U = 4,5$ V, $T_{\text{měření}} = 10$ m = 600 s								

Bylo simulováno několik scénářů použití zařízení, podle počtu účastníků, kteří dorazí na stanoviště. Žádný, jeden, tři, pět a deset. Měření pro tři a pět účastníků proběhlo dvakrát – pro případ postupného a hromadného příchodu. Pro každé měření je stanoven průměrný, min a max proud. Dále je vypočtena doba vysílání. Konkrétně se jedná o množství času, kdy odebíraný proud je větší než 5 mA. Následně, při rozdělení měření na spánek a aktivitu, byl stanoven průměrný proud při těchto režimech. Z těchto hodnot byl vypočten průměrný odebíraný výkon, při vysílání a při režimu spánku podle vzorce 4.2

$$P_{avg} = I_{avg} \cdot U \quad (4.2)$$

Ze získaných hodnot byly stanoveny energie podle vzorce 4.3 pro oba režimy a z těchto dvou hodnot byla stanovena celková energie na scénář podle vzorce 4.4

$$E = P_{avg} \cdot t \quad (4.3)$$

$$E = E_{vysílání} + E_{spánek} \quad (4.4)$$

V tabulce v jednotlivých sloupcích jsou uvedeny hodnoty pro daný scénář. Podle očekávání, v případě, kdy na stanoviště nedorazí žádný účastník, je spotřebovaná energie nejmenší a naopak při deseti účastnících největší. I v případě dvou měření pro tři nebo pro pět účastníků je celková spotřebovaná energie podobná. Jejich malé rozdíly mohou být způsobeny lehce odlišným průběhem měření. Při měření je totiž významné, v který čas se přiloží RFID čip ke čtečce. Jedná se o rozmezí až pěti sekund, kdy čtečka očekává přiložení čipu a po tu dobu je odběr vysoký.

4.3 Odhad výdrže na baterii

Výdrž baterie závisí na její kapacitě a na energii, která je z ní odebírána. Při provozu má velký vliv na výdrž i stárnutí baterií, nebo jejich samovolné vybíjení a okolní podmínky (teplota). Pro odhad výdrže pro vyvíjené zařízení byla použita AAA baterie Varta [32] s nominálním napětím článku 1,5 V a kapacitou 1 270 mAh. Je předpokládáno reálné využití 90 % kapacity (1 143 mAh). V zařízení jsou umístěny tři tyto baterie. Což podle vzorce 4.5 odpovídá 5,1435 Wh = 18 516 600 mWs.

$$Výdrž_{baterie} = kapacita \cdot U \quad (4.5)$$

Podle vzorce 4.6 byl vypočten teoretický maximální počet cyklů pro dané baterie. A podle vzorce byl stanoven 4.7 maximální čas provozu.

$$Počet_{cyklů} = \frac{E_{baterie}}{E_{celková}} \quad (4.6)$$

$$Čas_{provozu} = Počet_{cyklů} \cdot délka_{měření} \quad (4.7)$$

Tabulka 4.3: Výdrž baterie v různých scénářích použití

	0	1	3	3 _h	5	5 _h	10
E_celková [mWs]	7182,00	7879,51	8930,15	9019,40	9778,97	9738,66	11804,52
Počet_cyklů	2578,19	2349,97	2073,49	2052,97	1893,51	1901,35	1568,60
Výdrž_baterie [dny]	17,90	16,32	14,39	14,26	13,15	13,20	10,89

V tabulce č. 4.3 je uvedena výdrž baterie pro určitý scénář, v případě, že by se tento scénář opakoval pořád dokola. V případě, že zařízení neobslouží ani jednoho závodníka (bude pouze v režimu spánku), by jeho výdrž měla dosahovat necelých 18 dní. V případě, že na stanoviště přijde průměrně jeden účastník za jednu minutu, se výdrž baterie pohybuje okolo jedenácti dní. Tudíž tyto baterie jsou dostačující i pro vícedenní akci.

Kapitola 5

Návrh a implementace SW aplikace pro zpracování a vizualizaci dat o závodnících

Při navrhování jakékoli aplikace je třeba mít celou dobu na paměti jednu zásadní věc. A tou je, aby byl její celkový dojem takzvaně „user friendly“ (v doslovném překladu do češtiny toto spojení znamená „uživatelsky přátelský“). Dalo by se říct, že tato vlastnost pomáhá uživateli, aby se v celé aplikaci dobře orientoval. Nemělo by tak docházet například ke stavům, kdy uživatel neví, v jaké části se v aplikaci nachází, co po něm daná stránka požaduje, nebo jaké hodnoty má zadat, aby vykonal to, co v danou chvíli potřebuje.

Je potřeba myslet na skutečnost, že uživatel může být jen málo nebo průměrně znalý v oblasti informatiky a nemá tudíž srovnatelné intuitivní reagování jako třeba člověk, který aplikaci vyvíjí. Dalo by se říct, že tato fáze zasahuje až do oblasti psychologie, kdy je třeba dobře odhadnout reakce jednotlivých uživatelů a tomu odpovídající chování programu.

Při návrhu aplikace bylo počítáno s tím, že v budoucnu by tento systém mohly použít další organizace, které pořádají podobné závody. Proto byl na tuto část kladen větší důraz, hlavně z důvodu jednoduššího používání celého systému.

Aplikace byla vyvíjena pro konkrétní závod. Pro objasnění jednotlivých pojmů, je zde uveden průběh a pravidla tohoto závodu. Z těchto informací jsou vyvozeny požadavky na aplikaci. Následně jsou popsány použité technologie pro tvorbu aplikace. Samotnou aplikaci je možné rozdělit na tři části. První částí je správa dat před a po závodu, druhou částí je zpracování dat a výpočet výsledků a pomocí třetí části je možné sesbíraná data vizualizovat. Tyto části jsou v následujícím textu více popsány.

5.1 Pravidla a průběh hry

Tento systém byl vytvářen pro hru Copak je to za vojáka. Je to skautská akční hra na motivy stejnojmenného filmu. Úkolem účastníků je dostat ze startu do cíle v co nejkratším čase. Herní území se rozkládá na ploše zhruba 20 km². Účastníci se mohou po tomto území libovolně pohybovat. Na začátku má každý účastník čtyři životy, o které může přijít (strhnutím pásky), když narazí na vojenskou policii. Za každý ztracený život se k celkovému času přičte 45 minut. Na předem známých místech jsou kontrolní stanoviště, na která účastníci však nemusí dojít, ale přijdou tak o možnost získat časový bonus. Ukázka mapy herního území je v příloze na obrázku č. A.2. Vítězem se stává účastník s nejkratším výsledným časem. Tento čas se vypočítá z celkového času stráveného na trati a započítání bonusů za stanoviště. Stanoviště na trase jsou rozdělena do dvou kategorií – probíhací a úkolová.

Probíhací stanoviště – při doběhnutí na místo účastník přiloží čip k zařízení a následně může pokračovat dál. Na těchto stanovištích se na čip zapíše číslo stanice a aktuální čas. Při závěrečném vyhodnocení se v takovém případě účastníkovi odečte 20 minut z celkového času. Tato stanoviště jsou rozmístěna po trase ve skupinách po třech. Na mapě (obrázek č. A.2) je jedna skupina označena např. „RUN 1a, RUN 1b, RUN 1c“. Pro účastníky to znamená, že si mohou vybrat jen jedno stanoviště ze skupiny, na které poběží. Pokud doběhnou na dvě stanoviště z jedné skupiny, započítá se jim jen jeden časový bonus.

Dále jsou na trase **úkolová stanoviště**. Při doběhnutí na toto stanoviště účastník přiloží čip k zařízení pro zápisu času příchodu. Poté má na výběr. Buď poběží dál, tudíž se mu odečte jen 20 minut stejně jako u probíhajícího stanoviště, nebo splní úkol, který je na daném místě připraven (střelba na terče, zdolat horolezeckou stěnu...). Za splnění úkolu může získat až 30 bodů = 30 minut časový bonus. Před opuštěním stanoviště účastník znovu přiloží čip k zařízení pro zápis času odchodu. Díky tomu známe čas, který účastník stráví na stanovišti, a je ho možné odečíst do výsledného času. Pro lepší přehlednost je uvedena tabulka bodování.

Tabulka 5.1: Přehled bodování

Kde	Co	Změna celkového času
probíhací stanoviště	průchod 2. ve stejné skupině	- 20 minut 0
úkolová stanoviště	průchod úkol čekací čas	- 20 minut až - 30 minut - čas na stanovišti
hrací pole	ztráta života	+ 45 minut

5.1.1 Požadavky na aplikaci

Níže jsou uvedeny požadavky na vyhodnocovací systém. Tento systém bude obsahovat databázi a zajistí funkce pro příjem a zpracování dat a následně vyhodnocení výsledků závodu. Jednotlivé požadavky na aplikaci jsou:

- data z RFID čipů zapíše do databáze,
- data zaslaná přes LoRaWAN zapíše do databáze,
- umožní uživateli zapisovat data do tabulek,
- zajistí výpočet výsledků, konkrétně:
 - vypočte celkový čas účastníka na trati (cíl – start),
 - odečte 20 minut za návštěvu probíhajícího stanoviště,
 - odečte čas strávený na úkolovém stanovišti,
 - odečte čas přepočítaný z bodů na úkolovém stanovišti (maximálně 30 bodů, 1 bod = 1 minuta),
 - přičte čas za ztracené životy (1 ztracený život = 45 minut),
- poskytne kontrolu správnosti dat,
- vizualizuje výsledky závodu.

5.2 Použité technologie

Následuje výpis použitých technologií. Celý systém běží na Raspberry Pi, na kterém je nainstalován (OS) operační systém Raspbian. Jedná se o 32bitový systém odvozený z Debianu (Linux). Raspbian je vysoce optimalizovaný pro ARM procesory užívané v Raspberry Pi.

Pro hlavní část aplikace byl použit balíček LAMP. Jedná se o skupinu technologií vhodnou pro budování dynamických webových stránek a webových aplikací. Je to zkratka pro Linux, Apache, MariaDB a PHP. Společnou vlastností všech čtyř technologií je, že se jedná o volně šiřitelný software.

Raspberry Pi

Pro tuto práci byl použit RPi 4 Model B – 4 GB RAM. Tento minipočítač obsahuje 64bitový procesor Broadcom BCM2711 s taktem 1,5 GHz architektury ARM, 4 GB paměti RAM, 4x USB port, 1x 100 Mbit/s Ethernet a WiFi, HDMI. Operační systém a data jsou uložena na microSD kartě. Napájený je pomocí USB.

Webový server Apache

Dále je potřeba na RPi nainstalovat obslužnou aplikaci, která zajistí základní funkčnosti serveru. Tím bude v našem případě webový server s názvem Apache 2.

HTML

Hypertext Markup Language neboli značkovací jazyk, využívající hypertextové odkazy se používá pro tvorbu webových stránek. V HTML souboru je dána struktura a obsah webové stránky užitím takzvaných elementů.

CSS

Kaskádové styly (v anglickém jazyce Cascading Style Sheets) zkráceně CSS je technologie, která odděluje obsah webové stránky od vzhledu. Pomocí CSS lze jednoduše definovat druhy písma, barvu pozadí stránky, zarovnání, rozlišení a další vlastnosti elementů. Výhoda kaskádových stylů je, že tyto deklarované vlastnosti mohou sdílet všechny stránky celého webu.

PHP

PHP (Hypertext Preprocessor) je skriptovací programovací jazyk. Je určen především pro programování dynamických internetových stránek a webových aplikací. Při použití PHP pro dynamické stránky jsou skripty prováděny na straně serveru – k uživateli je přenášén až výsledek jejich činnosti.

JavaScript

JavaScript (JS) je multiplatformní, objektově orientovaný, událostmi řízený skriptovací jazyk. Vkládá se přímo jako součást HTML kódu stránky. Interpretaci v tomto případě provádí webový prohlížeč návštěvníka stránky. Jsou jím obvykle ovládány různé interaktivní prvky GUI (tlačítka, textová políčka) nebo tvořeny animace a efekty obrázků. Prakticky to znamená, že se program v JavaScriptu obvykle spouští až po stažení webové stránky z Internetu (tzv. na straně klienta), na rozdíl od ostatních jiných interpretovaných programovacích jazyků (např. PHP a ASP), které se spouštějí na straně serveru ještě před stažením z Internetu.

Bootstrap

Bootstrap je svobodná a otevřená sada nástrojů kaskádových stylů pro tvorbu webu a webových aplikací. Obsahuje návrhářské šablony založené na HTML a CSS, sloužící pro úpravu typografie, formulářů, tlačítek, navigace a dalších komponent rozhraní, stejně jako další volitelná rozšíření programovacího jazyka JavaScriptu.

MarriaDB a SQL

Databáze MarriaDB je známá pod názvem MySQL, ale z důvodu prodeje společnosti Oracle Corporation a změně v politice vydávání nových verzí se skupina původních vývojářů rozhodla odštěpit a na poslední veřejné verzi postavit nový databázový systém pojmenovaný MarriaDB. Nynější verze

MarriaDB je mnohem aktivněji vyvíjena než současná verze MySQL. Jedná se o klasickou databázi často používanou webovými aplikacemi uplatňující relační databázový model.

Pro práci s databázemi na bázi tabulek se používá jazyk Structured Query Language (SQL). Jedná se o jazyk vyvinutý pro dotazování do databáze, aby se co nejvíce podobal přirozenému anglickému jazyku. Mimo samotné dotazování obsahuje příkazy i pro ukládání a změnu dat, správu samotné databáze a práce se strukturou tabulek.

Node-RED

Node-RED je programovací nástroj pro propojení hardwarových zařízení, API a online služeb jako součást internetu věcí. Node-RED poskytuje editor pro vizuální programování ve webovém prohlížeči. K programování aplikací se využívá tzv. flow-based přístup. Ten je založen na tzv. uzlech (nodes), z nichž každý plní specifickou funkci, podle které zpracovává příchozí data a odesílá je do dalšího uzlu v pořadí. Samotné chování aplikace pak je tvořeno prostým umístováním a propojováním uzlů do tzv. toku (flow). V úložišti je paleta obsahující více než 225 000 uzlů.

Web Serial API

Serial API poskytuje webovým serverům možnost čtení a zápisu ze sériového zařízení pomocí skriptu. Takové API propojuje internet a fyzický svět tím, že umožňuje vzdálenému serveru komunikovat se zařízeními, jako jsou mikrokontrolery, 3D tiskárny a další sériová zařízení. Tato možnost je zajištěna díky JavaScriptu, který běží lokálně u uživatele v prohlížeči. Je možné se připojit k USB portu na daném počítači a číst případně zapisovat na něj data. Web Serial API je asynchronní. Díky tomu se nemůže stát, že požadavek na čtení bude blokovat celou aplikaci při čekání na vstup.

R Shiny Server

Shiny Server je back-end open source program. Vytváří webový server speciálně navržen pro hostování shiny R aplikací. Aplikace je pak možné zpřístupnit na internetu. Shiny Server hostuje každou aplikaci na její vlastní webové adrese a automaticky ji spustí, když uživatel danou adresu navštíví. Když uživatel odejde, Shiny server aplikaci automaticky zastaví. Ještě před samotnou instalací softwaru Shiny Server je nutné mít nainstalován software R a také balík Shiny.

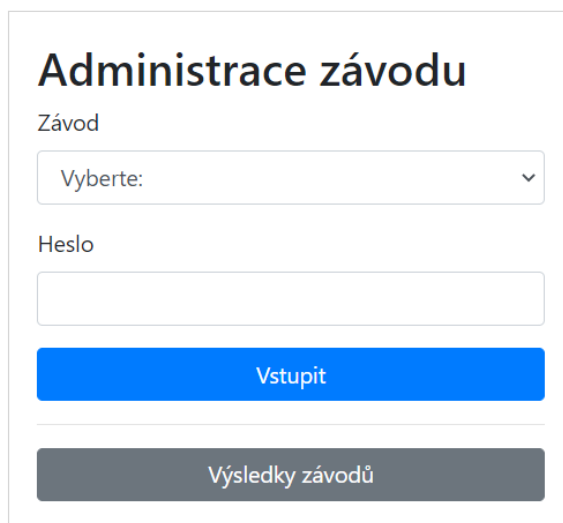
5.3 Sběr a administrace dat

Tato část popisuje aplikaci pro editaci dat. Data pro každý závod jsou přístupná pouze po zadání hesla. Všechna data jsou uložena v databázi. Pro každý nový závod je potřeba zadat určité údaje do systému ještě před startem závodu, zapsat data v průběhu závodu (často pomocí mobilního zařízení) a po závodu je potřeba načíst a uložit data z RFID čipů účastníků. Následuje popis jednotlivých formulářů a načítání dat.

5.3.1 Přihlášení

Při zobrazení první stránky se načte přihlašovací formulář, který je možné vidět obrázku č. 5.1. Po zadání hesla je možno editovat údaje pro daný závod nebo při stisku tlačítka „Výsledky závodů“ zobrazit dashboard s výsledky, který je popsán dále v kapitole 5.5.

Po správně vyplněných položkách závod a heslo a následném stisku tlačítka „Vstoupit“ se tyto přihlašovací údaje uloží do Cookies. Takže například, když už uživatel vstoupil do editace daného závodu, tak při načtení úvodní stránky je rovnou přesměrován na editaci závodu. V případě potřeby se na hlavní obrazovce nachází funkce pro odhlášení.



Administrace závodu

Závod

Vyberte: ▾

Heslo

Vstoupit

Výsledky závodů

Obrázek 5.1: První stránka – přihlášení

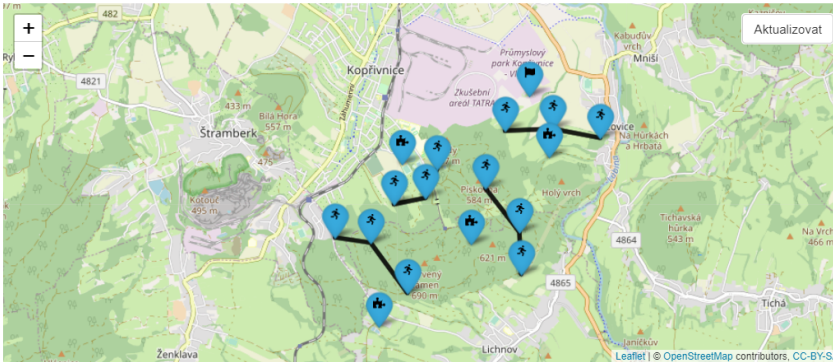
5.3.2 Editace záznamů

Data, která jsou editovaná manuálně, je možné rozdělit do dvou kategorií. Jednak jsou to data, která je potřeba zadat před závodem – jedná se hlavně o údaje o závodnících a stanovištích, a dále v průběhu závodu, se budou zapisovat body, které získají účastníci za splnění úkolů na stanovištích.

Údaje před závodem

Ještě než závodníci odstartují, je potřeba mít o nich záznamy v systému. To samé platí i v případě umístění stanovišť. Jako ukázka je na obrázku č. 5.3 zobrazena tabulka stanovišť. U této tabulky je navíc vykresleno umístění uvedených stanovišť na mapě.

Stanoviste







A map of the Stramberk area showing the locations of various stations. The map includes labels for Stramberk, Kopřivnice, and surrounding hills. Blue pins mark the locations of the stations. Below the map are buttons for 'Editace' and 'Export do excelu'.

ID	Nazev	Ukolove	ID_Skupina	Aktivni	GPSN	GPSE	Popis	Poradi
1	START	2	0	1	49.59569690	18.17468060	CP Start	1
10	CP_01	1	0	1	49.58809890	18.17837920	Airsoft	3
11	RUN_01B	0	1	1	49.59154890	18.17915170	RUN_01A	2
12	RUN_01A	0	1	1	49.59138190	18.17005360	RUN_01B	2
13	RUN_01C	0	1	1	49.59043580	18.18824970	RUN_01C	2
20	CP_02	1	0	1	49.57735860	18.16344470	Zprava	5
21	RUN_02A	0	2	1	49.58425940	18.16619140		4

Obrázek 5.2: Zobrazení údajů – stanoviště

U každé tabulky je pak možnost data editovat, případně exportovat a uložit v počítači. Po kliknutí na položku „Editovat“, se načte zobrazení pro editaci. Ukázka je na obrázku č. 5.3.

Import Odstranit vše Ukončit editaci

ID	Nazev	Ukolove	ID_Skupina	Aktivni	GPSN	GPSE	Popis	Poradi	Akce
1	START	2	0	1	49.59569690	18.17468060	CP Start	1	 
10	CP_01	1	0	1	49.58809890	18.17837920	Airsoft	3	 

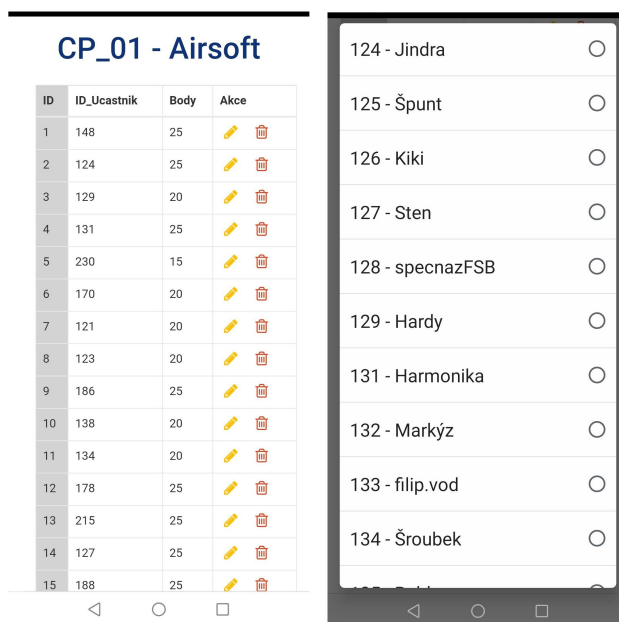
Obrázek 5.3: Editace údajů

V editačním zobrazení se změnila nabídka nástrojů. Po změně jsou dostupná tlačítka pro import dat, odstranění všech záznamů v tabulce a ukončení editace. Dále se na každém řádku objeví tlačítko pro editaci nebo pro smazání celého řádku. Pod tabulkou se zobrazí tlačítko pro přidání nového řádku. Když uživatel provede některou akci, ihned se to projeví ve vizualizované tabulce a v databázi.

Zadávání bodů

V případě zápisů bodů z úkolových stanovišť byl při vývoji aplikace kladen důraz na vhodné zobrazení na malé obrazovce mobilního telefonu. Organizátoři na úkolových stanovištích budou moci zapsat body za splnění úkolu rovnou do systému, s předpokladem, že mají mobil s internetovým připojením. Na obrázku 5.4 je možné vidět snímky obrazovky z mobilního zařízení. Vlevo je zobrazena tabulka s již vloženými záznamy. V tomto případě jsou důležité dva údaje. ID účastníka, které má na průvodce, a počet bodů za splněný úkol. Při vkládání nových záznamů, případně při editaci se pole pro zadání ID účastníka změní na výběrový seznam, který pomůže zapisovateli ve výběru správného závodníka. Náhled je možné vidět na obrázku vpravo. Každá položka se skládá z ID a přezdívkou účastníka.

Podobným způsobem je možné zapsat i ztracené životy.



Obrázek 5.4: Sběr dat – zobrazení na mobilním telefonu

5.3.3 Společný start

Mezi užitečné funkce, které aplikace obsahuje, patří funkce Společný start. Tuto funkci je vhodné použít v situaci, jako například v případě výše popsaného závodu, kdy všichni účastníci startují závod ve stejnou dobu, případně startují po skupinách. V takovém případě je nesmyslné, aby si všichni účastníci zapisovali čas startu pomocí zařízení na RFID čip, ale je vhodné jej zapsat do databáze přímo v systému, což právě umožňuje tato funkce.

Vedlejší použití této funkce může být v případě nefunkčnosti některých zařízení na trase. V případě, když účastník objeví nefunkční zařízení, má možnost zavolat na základnu a sdělit jim tuto

informaci. Následně organizátor má možnost zapsat, pomocí této aplikace, průchod účastníka daným stanovištěm přímo do systému.

Při použití této funkce je tedy potřeba vybrat účastníky, buď všechny nebo pomocí zaškrtnutí políčka vybrat konkrétní osoby, dále nastavit čas „průchodu“ a vybrat ze seznamu dané stanoviště.

5.3.4 Data na RFID čípech

Problematikou v tomto případě bylo najít způsob, jak načíst data o průchodech, zapsané na RFID čípech, do databáze bez toho, aby uživatel musel stahovat/installovat aplikaci. Pro tyto účely slouží Web Serial API, která dokáže číst data z USB portu.

Komunikace USB

Připojeno k USB

Data o průchodech

Zápis ID čipu

Aktualizace času

Po přiložení čipu načte záznamy o průchodech.

☒ Spustit čtení

Možnosti:

☐ Automaticky zapisovat do databáze

☐ Po přečtení smazat

ID čipu	ID stanoviště	Den	Hodiny	Minuty	Sekundy
103	100	22	11	25	51
103	20	22	17	14	44
103	20	22	17	5	47
103	11	22	16	50	43
103	10	22	16	40	2
103	10	22	16	33	50

Obrázek 5.5: Komunikace s RFID čipy

Při použití tohoto API je na uživatele kladen jediný požadavek – použít prohlížeč Google Chrome a povolit v něm experimentální funkce. Bez splnění těchto podmínek není možné navázat spojení s USB portem a uživateli se vypíše chybová hláška. Pro vývojáře je požadavek na zabezpečené HTTPS (Hypertext Transfer Protocol Secure) spojení. Dále je potřeba zařízení s RFID anténou a USB konektorem, které data vyšle po sériové lince.

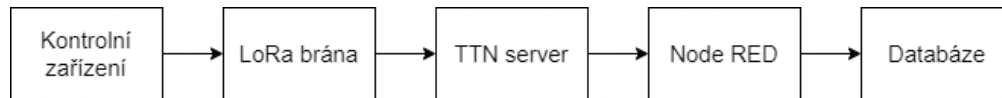
Na obrázku č. 5.5 je ukázka aplikace. Nejprve vyzve uživatele k výběru sériového portu a následně se k němu připojí a po přiložení čipu ke čtečce načte data do tabulky. Uživatel si pak může vybrat, jestli chce údaje uložit do databáze a případně z čipu rovnou smazat.

Dále se v této aplikaci nachází záložka pro aktualizaci času v zařízení a zápisu ID na RFID čipy, které fungují obdobným způsobem jako čtení dat.

5.3.5 Data zaslané pomocí LoRaWAN

Díky datům posílaným přes LoRaWAN mohou mít organizátoři přehled o aktuálním stavu hry. Mohou si na mapě zobrazit, kde se přibližně účastníci pohybují, zjistit aktuální stav baterie v zařízení nebo teplotu uvnitř zařízení.

Způsob přenosu dat ze zařízení do databáze, je znázorněn v následujícím schématu č. 5.6.



Obrázek 5.6: Schéma toku dat LoRa

V případě kontrolních zařízení je potřeba, aby jejich kód obsahoval parametry vygenerované serverem TTN. Následně, když bude zařízení v dosahu LoRa brány, která je připojená k síti TTN, může vyslat data, která pak se zobrazí v konzoli TTN u daného zařízení. Na serveru je pak možné pro celou aplikaci (skupinu zařízení) nastavit formát posílaných dat.

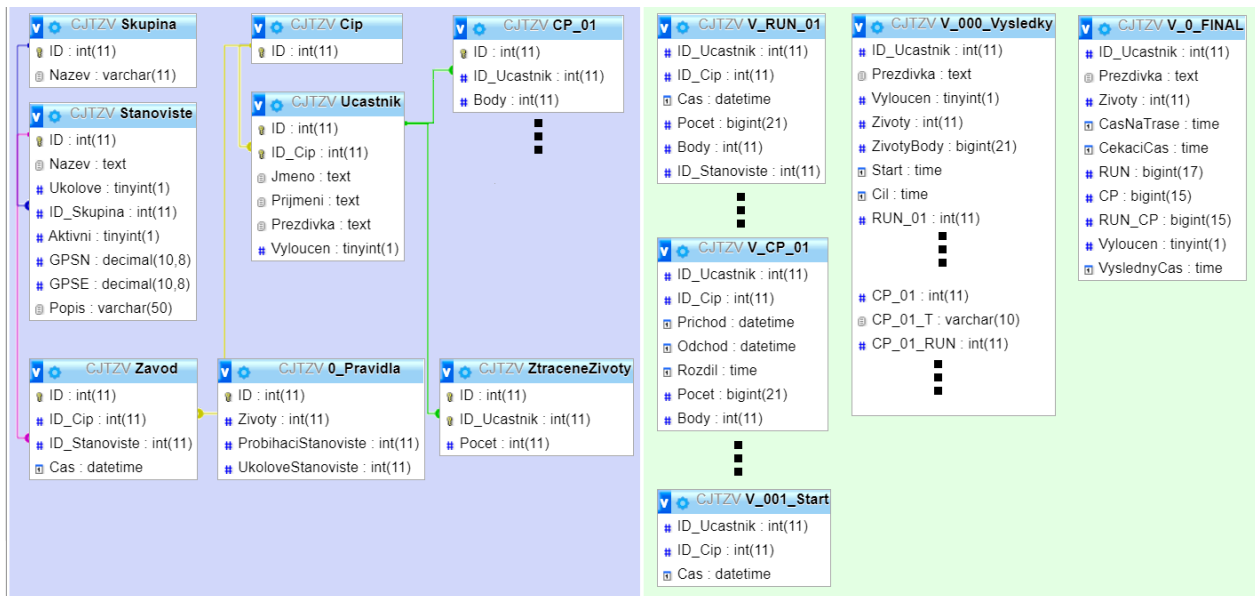
Příchozí data se na TTN serveru nearchivují. Možné je pouze zobrazit příchozí data v době otevřeného náhledu dat. Po aktualizaci stránky se list vymaže a opět budou k dispozici pouze nově příchozí data. TTN server ale umožňuje přeposílání dat různými způsoby. Nastavené přeposílání dat vždy platí pro veškerá zařízení, v jedné aplikaci.

Jednou z možností, jak data dále zpracovat, je pomocí Node-RED. Příchozí data se jednak zapíší do souboru pro případné zpětné dohledání. A pomocí bloku pro komunikaci s databází a SQL příkazem INSERT se data uloží do databáze. Pro lepší přehled o přijatých datech je vytvořen jednoduchý dashboard, kde je možné vidět poslední přijatá data, čas jejich přijetí a čas posledního zápisu do databáze.

5.4 Zpracování dat a výpočet výsledků

Celý systém je navržen tak, aby bylo možné zpětně analyzovat data z předchozích ročníků a aby bylo možno systém snadno znovu použít na další závod. Je vytvořena jedna vzorová databáze, kdy pro novou akci se vždy vytvoří její kopie. Takže pro každou akci je vytvořena nová databáze, která má stejnou strukturu. Následuje popis jedné takové databáze. Data poslána přes LoRaWAN se vždy ukládají do další, pro všechny akce společné databáze.

Databáze se skládá z několika tabulek a dotazů. Na obrázku 5.7 je uvedeno schéma databáze. Schéma je rozděleno na dvě části: tabulky (vpravo) a dotazy (vlevo). U některých objektů jsou zakresleny tři tečky. Toto znázorňuje vícenásobné použití stejné struktury tabulky. Jedná se například o tabulky pro zápis bodů z jednotlivých úkolových stanovišť. Následuje popis jednotlivých objektů.



Obrázek 5.7: Schéma databáze

Mezi nejdůležitější tabulky, které je potřeba vyplnit před startem závodu, patří tabulka **Stanovisti**. Tato tabulka obsahuje *ID* jako primární klíč. Je také možné do tabulky napsat *Nazev* stanoviště. Položka *Ukolova* umožňuje určit, zda se jedná o úkolové nebo probíhací stanoviště. V případě, že se jedná o probíhací, je možné jako další hodnotu nastavit *ID_Skupina*. Tento sloupec je propojen s tabulkou **Skupina**. Slouží pro vytvoření skupin probíhacích stanovišť viz pravidla hry (kap. 5.1). Dále je možné nastavit souřadnice stanoviště (*GPSN* a *GPSE*) ve stupních – dvou desetinných číslech. Poslední položkou je *Popis*. Ten je vhodné použít jako název úkolu v případě úkolového stanoviště.

Druhou důležitou je tabulka **Účastník** sloužící pro uložení informací o účastnících. Jednou z položek této tabulky je i *ID_Cip*, která je provázána s tabulkou **Cip**, v níž se nachází ID aktivních

čipů. Mezi další položky patří *Jmeno*, *Prijmeni* a *Prezdivka*. V případě, že závodník se na závody přihlásí, ale nakonec se jich nezúčastní, je možné zaznamenat pomocí hodnoty *Vyloucen*.

Pomocí tabulky **Pravidla** jsou zapsány důležité konstanty pro vyhodnocení závodu. Jedná se o body, které se odečtou například při ztrátě života či proběhnutí stanoviště. Pro každý závod se budou tyto konstanty lišit.

Následující tabulky jsou plněny až v průběhu nebo na konci závodu. Mezi tyto tabulky patří tabulka **Závod**. Do této tabulky jsou po závodu načteny data o průchodech z RFID čipů. Konkrétně se jedná o položku *ID_cip*, na kterém byla data zapsána, navštívená *ID_Stanoviste* a *Cas* průchodu.

Mezi další průběžně plněcí se tabulky patří skupina tabulek **CP_***. Tyto tabulky slouží pro zápis bodů z úkolových stanovišť. Počet těchto tabulek závisí na obsahu tabulky Stanoviště – pro každé úkolové stanoviště se vytvoří jedna tabulka. Jednotlivé tabulky obsahují položky *ID_Ucastnik*, která je provázaná s tabulkou účastník, a počet *Body*, které účastník získal.

Podobnou strukturu má tabulka s názvem **ZtraceneZivoty**, slouží pro zápis počtu životů, o které účastník přišel v průběhu závodu. Obsahuje položku *ID_Ucastnik* a položku *Pocet* pro zápis počtu ztracených životů. V případě, že účastník neztratil ani jeden život, je možné buď zapsat nula bodů nebo nezapisovat vůbec.

Dále jsou vytvořené dotazy sloužící pro výpočet výsledků. Dotazy ze skupiny **V_RUN_*** slouží k výpočtu bonusových bodů za návštěvu probíhacích stanovišť. Hlavní problematika spočívá v tom, že probíhací stanoviště jsou rozdělena do skupin, kdy v každé skupině jsou přibližně tři stanoviště. Účastník by si měl vybrat vždy nanejvýš jedno stanoviště z dané skupiny. V případě, že navštíví více stanovišť z jedné skupiny, se mu započítá návštěva jen z jednoho stanoviště. Každý dotaz zastupuje jednu skupinu. Výsledkem tohoto dotazu je počet bonusových bodů pro dané účastníky.

Druhá skupina dotazů **V_CP_*** slouží pro výpočet času, který účastníci stráví na úkolových stanovištích. V případě, že účastník doběhne na úkolové stanoviště, má dvě možnosti, co bude dělat. Buď si označí jen příchod a poběží dál, tudíž dostane jen body jako za probíhací stanoviště. Nebo na stanovišti zůstane a splní nachystaný úkol. Následně je potřeba, aby zapsal na čip i odchod. V takovém případě dostane opět body za průchod jako v předchozím případě, ale má navíc body za splněný úkol a čas, který strávil na stanovišti, nebude započítán do celkového času.

Následně je možné všechny tyto hodnoty si zobrazit v přehledném dotazu **V_000_Vysledky** obsahující všechny bodované položky.

Pro finální výpočet výsledků slouží dotaz **V_0_FINAL**, kde jsou sečteny jednotlivé položky po kategoriích a je vypočten *Výsledný čas*, pomocí kterého se určí pořadí výherců závodu. Tento výsledný čas je reálný čas běhu účastníka – to znamená rozdíl času start a cíl, od toho je odečten čas, který strávil na úkolových stanovištích, dále jsou přičteny bonusové body jednak za návštěvu probíhacích stanovišť a za splnění úkolu na úkolovém stanovišti a přičtou se záporné body, za ztracené životy. Nakonec se účastníci rozdělí do kategorií podle počtu ztracených životů. Vyhrává hráč s nejkratším výsledným časem.

Oba poslední uvedené dotazy je možné v aplikaci zobrazit. Na obrázku č. 5.8 je ukázka zobrazení finálních výsledků. Výsledková listina je rozdělena do dvou kategorií a výherci jsou označeni zlatou barvou.

Finalní výsledky

Živí

ID_Ucastnik	Prezdivka	ZtraceneZivoty	CasNaTrase	CekaciCas	RUN	CP	RUN_CP	VyslednyCas
175	Špunt	0	04:17:02	00:09:59	120	30	25	01:12:03
146	matejkasto	0	05:44:01	00:19:57	60	72	70	02:02:04
131	Harmonika	0	04:17:02	00:08:04	60	25	25	02:18:58
215	rofo	0	08:36:04	00:44:56	80	127	100	02:44:08
138	Vítr	0	04:34:03	00:22:00	20	20	25	03:07:03
121	Makovice	0	06:56:04	00:30:59	40	63	75	03:27:05

Mrtví

ID_Ucastnik	Prezdivka	ZtraceneZivoty	CasNaTrase	CekaciCas	RUN	CP	RUN_CP	VyslednyCas
192	stefek.ho	2	04:17:03	00:17:03	100	51	50	02:09:00
129	Hardy	1	06:10:03	01:33:01	0	65	75	03:02:02
167	Boban	3	04:17:02	00:19:06	100	24	50	03:18:56
178	Baghíra	1	04:34:04	00:28:01	20	25	25	03:41:03
134	Šroubek	1	04:34:03	00:24:59	20	20	25	03:49:04

Obrázek 5.8: Zpracované výsledky

V případě zobrazení podrobných výsledků je obsaženo více barevných odlišení. Díky tomuto zobrazení je možné ověřit, že někde v datech nenastala chyba. Příkladem chyby může být, že účastník byl označen jako vyloučený, ale přitom se závodu účastnil. Dále se může stát, že účastník, který splnil úkol na úkolovém stanovišti, má zapsaný čas, který na stanovišti strávil, ale nemá zapsané body, případně naopak. Tyto chyby budou v tabulce zvýrazněny a pro správné vyhodnocení závodu je potřeba je opravit.

5.5 Vizualizace dat

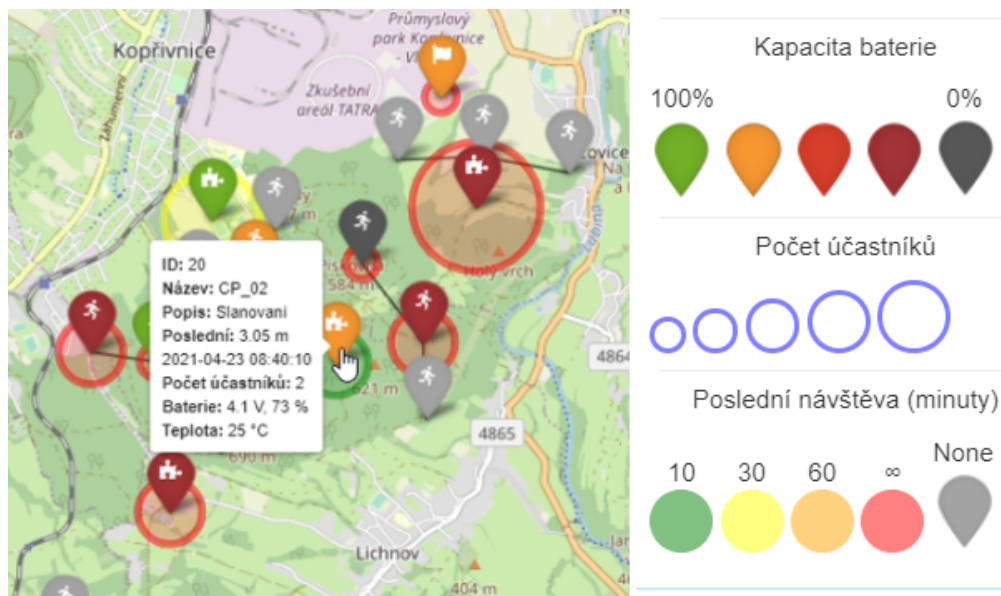
Vizualizace dat jsou tvořené pomocí jazyka R a knihovny Shiny, která umožňuje vytvářet webové aplikace, tudíž grafy je možné sledovat online.

Tyto aplikace běží na Shiny serveru. Jsou vytvořeny tři aplikace. Mapa se stanovišti, online závod a přehledný dashboard s analýzou závodu. Mapa již byla zobrazena na snímku č.5.3 a další dvě jsou popsány níže.

5.5.1 Online závod

Zdrojem dat pro tuto vizualizaci jsou data zaslaná pomocí LoRaWAN – tedy data o aktuálním stavu hry. Ukázka této vizualizace je na obrázku č. 5.9.

Na této ukázce je možné na pravé straně vidět mapu a na levé části vysvětlení, co znamenají jednotlivé ikony a jejich zbarvení. Levá část obrázku je ukázka postranního panelu, který je možné si přímo zobrazit s mapou.



Obrázek 5.9: Vizualizace – online závod

Na mapě je možné vidět několik informací. Jsou zde zobrazené ikony značící umístění stanovišť. Symbol uvnitř ikony značí druh stanoviště 🚩 start /cíl, 🏃 probíhací a 🏠 úkolové stanoviště. Barva ikony informuje o kapacitě baterie od zelené přes oranžovou, červenou až černou. V případě, že ještě nejsou žádná dostupná data ze zařízení (neprošel přes něj žádný účastník), je ikona zbarvena šedě. Kolem ikony stanoviště je vykreslen kruh. Velikost tohoto kruhu značí počet účastníků, kteří již prošli přes stanoviště, a jeho barva symbolizuje čas poslední aktivity.

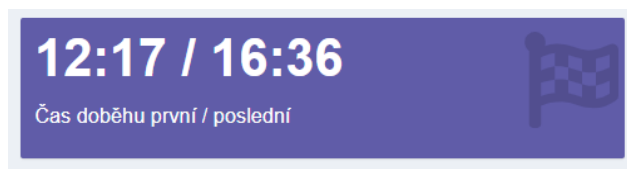
Při najetí kurzorem myši na ikonu stanoviště se zobrazí okno s podrobným popisem stavu. Okno obsahuje položky *ID*, *Název* a *Popis*, které identifikují stanoviště. Položka *Poslední* – určuje čas průchodu posledního závodníka. Je uveden rozdíl od aktuálního času a také daný čas průchodu. Položka *Počet účastníků* udává počet účastníků, kteří již od začátku závodu proběhli tímto stanovištěm. Dále jsou uvedeny položky *Baterie* a *Teplota*, které byly naměřeny v době posledního průchodu.

5.5.2 Dashboard - analýza závodu

Tento dashboard slouží pro přehlednou analýzu závodu třeba pro případ plánování dalšího ročníku hry. Obsahuje několik tematických záložek. Je možné se jednoduše přepnout na data z jiné akce. Následuje popis záložek, které dashboard obsahuje. U některých jsou uvedeny ukázky konkrétních grafů a vizualizací.

Přehled

Jako první se zobrazí stránka, která obsahuje několik boxů, v nichž jsou vypočteny určité hodnoty. Jako ukázka je na obrázku č. 5.10 zobrazen čas prvního a posledního účastníka v cíli.



Obrázek 5.10: Vizualizace – box s hodnotou

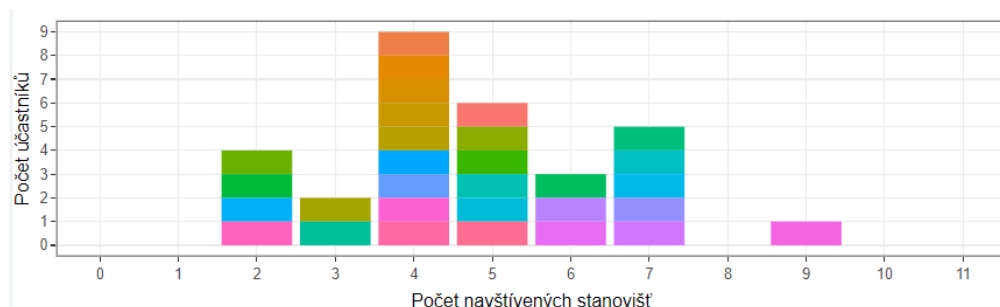
Mezi další zobrazené hodnoty patří například suma naběhaného času – to znamená součet všech časů, který účastníci stráví na trase, dále také délka trasy – vzdušná vzdálenost jednotlivých stanovišť.

Stanoviště

V záložce stanoviště jsou grafy, zobrazující počet účastníků na stanovištích. Na ukázku jsou zde uvedeny dva grafy. První na obrázku č. 5.11 ukazuje kolik účastníků navštívilo určitý počet stanovišť.

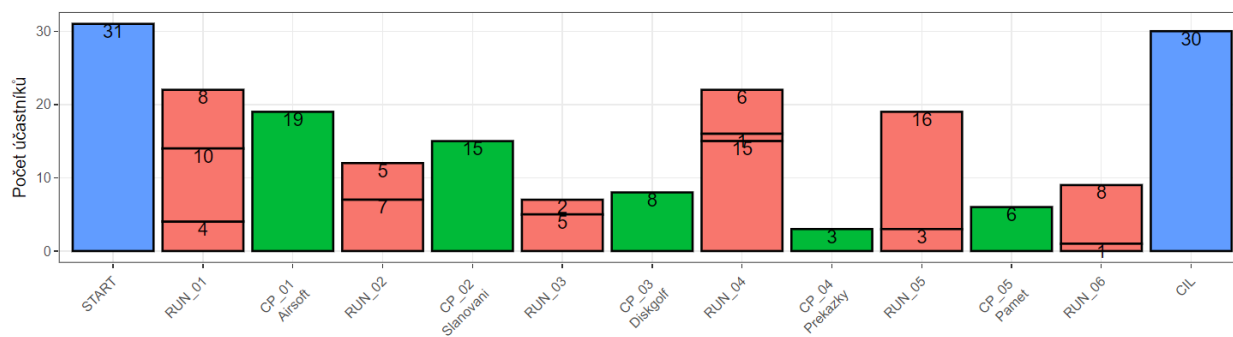
Pro vytvoření tohoto grafu bylo potřeba zjistit pro každého závodníka, kolik stanovišť navštívil a následně účastníky pomocí výsledného počtu seskupit – tato hodnota se nachází na ose x. Na ose y je pak počet účastníků v dané skupině. Z tohoto grafu je tedy možné vyčíst, že nejvíce závodníků navštívilo právě čtyři stanoviště. Do počtu průchodů stanovišti není započítán průchod startem a cílem.

Pomocí druhého grafu na obrázku č. 5.12 je možné zjistit, kolik účastníků bylo na daném stanovišti. Stanoviště jsou seřazena podle toho, v jakém pořadí jsou na mapě. Pomocí barev jsou odděleny typy stanovišť – modrá označuje start a cíl, oranžová značí probíhací a zelená úkolové stanoviště.



Obrázek 5.11: Vizualizace – navštívené stanoviště

V případě probíhacích (oranžových) sloupců je možné vidět, že jsou rozděleny na více menších částí z důvodu, že probíhací stanoviště jsou rozdělena do skupin. Tudiž jeden sloupec v grafu zobrazuje skupinu více stanovišť.



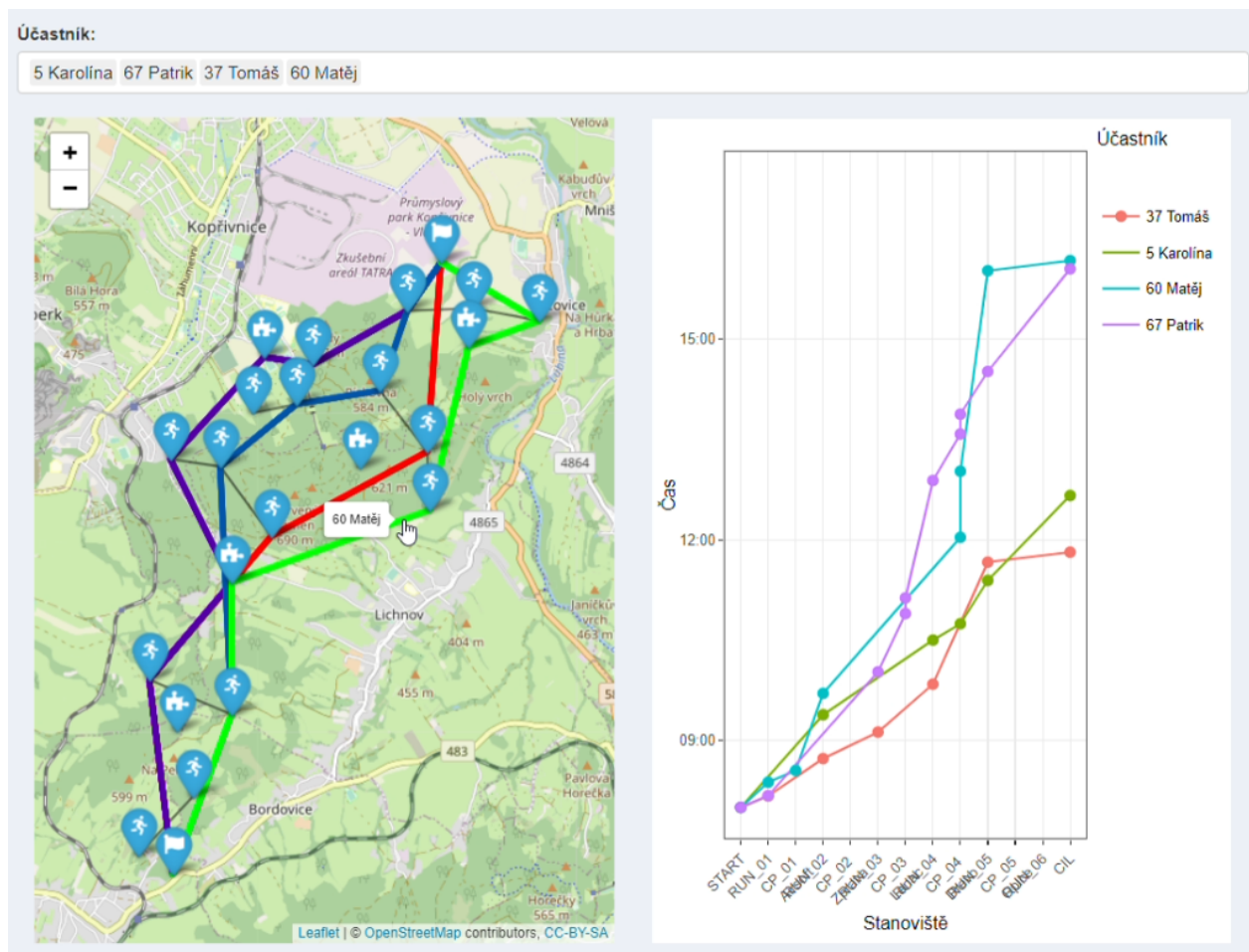
Obrázek 5.12: Vizualizace – účastníci na stanovišti

Účastníci

Na další záložce s názvem účastníci je možné si vykreslit trasu účastníků. Ukázka je na obrázku č. 5.13. Trasu je možné vykreslit jak pro konkrétní účastníky (výběr je možný podle ID a přezdívký), tak pro všechny účastníky. Na ukázce jsou vybraní čtyři závodníci. Výběrové okno je v horní části. Ve zbylé části jsou zobrazené dva grafy.

V levé části obrázku je vykreslena trasa v mapě. Opět je možné vidět různé symboly v ikonách označující stanoviště. Trasa každého účastníka je vykreslena jinou barvou. Jak je možné vidět, při najetí kurzorem na určitou čáru, zobrazí se informace, kterému účastníkovi tato trasa patří.

Ve druhém grafu (na pravé straně) je trasa vykreslena v závislosti na čase. Na x ose jsou jednotlivá stanoviště opět seřazená podle toho, v jakém pořadí jsou na mapě. Na y ose je čas průchodu.



Obrázek 5.13: Vizualizace – trasy závodníků

Simulace závodu

Poslední záložkou v tomto dashboardu je simulace průběhu závodu. Pomocí téhle simulace je možné se zpětně podívat na průběh závodu. Vizualizace vypadá podobně jako zobrazení online závodu jako na obrázku č. 5.9. Navíc obsahuje posuvník, pomocí kterého je možné ovládat čas, pro který se mají data vykreslit. Případně je možné spustit animaci, posuvník se pohybuje podle nastaveného kroku, například po 10 minutách. Pokaždé, při změně hodnoty posuvníku, se vykreslí nové objekty.

Kapitola 6

Zhodnocení dosažených výsledků

V rámci této práce byl realizován systém, který obsahuje pět kontrolních zařízení a webovou aplikaci. Pro zhodnocení dosažených výsledků bylo provedeno kompletní otestování systému. Ověření funkčnosti proběhlo ve dvou hlavních krocích. Všechny důležité funkce byly otestovány v domácích podmínkách a následně byl proveden testovací závod v terénu.

V domácích podmínkách byla otestována funkčnost zařízení a aplikace. Pro otestování zařízení bylo potřeba nahrát do něj firmware obsahující vygenerované klíče ze serveru TTN. Bylo ověřeno, že zařízení správně zapisuje průchody účastníku na RFID čipy a vysílá data pomocí LoRaWAN. Data z RFID čipů při konečném nahrávání do databáze byla správně načtena přes Web Serial API. Obě tyto skutečnosti byly ověřeny přímo v databázi.

Pro ověření správnosti výpočtu výsledků závodu byl proveden výpočet výsledků pro předchozí ročníky závodu. Do systému byly nahrány data již z uskutečněných závodů a navrhované výsledky z aplikace byly porovnány s výsledky daných závodů. Vypočtené výsledky se shodovaly. Z toho bylo usouzeno, že funkce databáze a SQL dotazů je správná.

Dalším krokem bylo testování webové aplikace. Byla vytvořena nová databáze pro testovací závod, vyzkoušeny zápisy do jednotlivých tabulek a vyzkoušena funkčnost vizualizací. Podařilo se úspěšně založit závod, vytvořit záznamy o účastnících a stanovištích. Tyto údaje byly použity pro zkušební závod.

V tomto závodě bylo použito pět stanovišť (pět zařízení). Závodu se zúčastnilo pět fiktivních závodníků a jejich průběh přes stanoviště byl nasimulován. Testování nemohlo být uskutečněno přímo v místě obvyklého závodu, neboť tento prostor zatím není pokrytý signálem LoRa. Testování probíhalo v Porubském lese nedaleko areálu VŠB-TUO, kde pokrytí signálem sítě Lora bylo dostačující. V tomto testovacím závodě bylo provedeno celkem 26 zápisů. Všechny tyto zápisy byly nalezeny na RFID čipech. Pomocí LoRaWAN byly zjištěny dva neúspěšné záznamy. To mohlo být způsobeno špatným signálem v daném místě.

Všechny testy proběhly úspěšně. Případné nedostatky byly ihned vyřešeny. Postup testování při zkušebním závodě je uveden v příloze č. B.

Kapitola 7

Závěr

Cílem této diplomové práce bylo vytvořit systém sloužící pro identifikaci závodníků a vyhodnocení výsledků orientačního závodu. Systém se skládá z kontrolních zařízení a aplikace pro správu a vizualizaci dat. Celý systém je navržen pro konkrétní závod, ale je ho možné použít pro jakýkoliv závod s podobnými pravidly.

Úkolem kontrolních zařízení je vytvořit záznam o průchodu účastníka přes stanoviště. Tato zařízení budou umístěna v hracím poli a v případě, že účastník doběhne na stanoviště, zařízení zapíšou tuto informaci na účastníkův RFID čip. Součástí navrhovaných zařízení je modul pro komunikaci pomocí LoRaWAN a je možné si zobrazit data o průběhu závodu v počítači v reálném čase. U těchto zařízení byly vybrány vhodné moduly a navrženo schéma zapojení a deska plošných spojů. Dále byl vytvořen firmware pro tato zařízení a v práci jsou popsány nejdůležitější části kódu. Zařízení jsou připojena k LoRaWAN v rámci projektu TTN. Vzhledem k tomu, že na herním území je nulové pokrytí signálem, byla vyjednána instalace LoRa brány na nedaleké rozhledně.

Realizovaná zařízení jsou napájena z baterií, z toho důvodu byla provedena analýza energetické spotřeby. Byly nasimulovány různé scénáře použití zařízení, podle počtu obslužených účastníků, a pro tyto scénáře byla stanovena výdrž baterie.

Druhou částí práce byl návrh a realizace webové aplikace pro administraci a vizualizaci dat. Aplikace je uživatelsky přívětivá. Byl kladen důraz na její jednoduché a intuitivní ovládání. Aplikace se všemi podpůrnými technologiemi byla nainstalována na serveru (Raspberry Pi). Aplikace se dělí na tři části. První část spravuje data před a po závodu. Druhá část zpracovává data a vypočítává výsledky. Třetí část umožňuje sesbíraná data vizualizovat. Pro administraci dat byly použity hlavně technologie HTML, PHP a JS. Tato část obsahuje funkce pro čtení a zápis dat do databáze. Navíc byla vytvořena webová služba pro čtení dat z RFID čipů. Pro zpracování dat a výpočet výsledků slouží MariaDB databáze. Všechny potřebné výpočty pro vyhodnocení závodu jsou prováděny pomocí SQL dotazů. Pro každou novou akci bude vytvořena nová databáze.

Vizualizace dat je vytvořena pomocí jazyka R a knihovny Shiny. V rámci této technologie jsou vytvořeny tři aplikace, které jsou spuštěny na Shiny Serveru na Raspberry Pi. Je vytvořena malá

aplikace, která zobrazí umístění stanovišť na mapě. Tato vizualizace se zobrazí při úpravě záznamů o stanovištích. Pomocí další Shiny aplikace je možné sledovat průběh závodu v reálném čase. Zdrojem dat pro tuto vizualizaci jsou data zasláná pomocí LoRaWAN. Na této vizualizaci můžeme vidět, kde se pohybují účastníci nebo stav baterie v zařízení. Třetí Shiny aplikací je přehledný dashboard všech dostupných dat. Tato vizualizace slouží pro plánování dalších ročníků akce. Pomocí ní je možné vykreslení trasy účastníků v mapě nebo v závislosti na čase. Nebo simulovat závod a zobrazit si pohyb účastníku v konkrétním čase.

V závěru této práce proběhlo kompletní testování celého systému. Ověření funkčnosti proběhlo ve dvou hlavních krocích. Všechny důležité funkce byly otestovány v domácích podmínkách a následně byl proveden testovací závod v terénu. Bylo zjištěno, že navrhovaný systém je funkční.

V rámci této práce se autorka blíže seznámila s návrhem desky plošných spojů a vývojem webové aplikace. Také si prohloubila znalosti jazyka R. Veškerý vývoj softwarové části probíhal za pomoci vývojového prostředí Visual Studio Code, případně RStudio a verzovacího systému Github.

V průběhu vývoje a testování bylo objeveno několik návrhů na vylepšení tohoto systému. Jedním z plánovaných vylepšení je zrychlení obsluhy závodníka (celý cyklus po stisknutí hlavního tlačítka). Bylo zjištěno, že při komunikaci s modulem RN2483 se spotřebovává hodně času. Jedním z možných řešení je použití jiné knihovny pro tento modul. Dalším návrhem je zvýšení počtu záznamů, které je možné na čip zapsat. Úpravou metody zápisu je možno zdvojnásobit paměť čipu. Případně použít čipy s větší pamětí.

Literatura

1. RAJI, R. S. Smart networks for control. *IEEE Spectrum*. 1994, roč. 31, č. 6, s. 49–55. Dostupné z DOI: 10.1109/6.284793.
2. *The Internet of Things* [online] [cit. 2020-12-02]. Dostupné z: <https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/the-internet-of-things>.
3. CULIC, Ioana; RADOVICI, Alexandru; RUSU, Cristian. Internet of Things Systems Overview. In: *Commercial and Industrial Internet of Things Applications with the Raspberry Pi: Prototyping IoT Solutions*. Berkeley, CA: Apress, 2020. ISBN 978-1-4842-5296-3. Dostupné z DOI: 10.1007/978-1-4842-5296-3_1.
4. AUGUSTIN, Aloÿs; YI, Jiazi; CLAUSEN, Thomas; TOWNSLEY, William. A Study of LoRa: Long Range and Low Power Networks for the Internet of Things. *Sensors*. 2016-09, roč. 16, č. 9, s. 1466. ISSN 1424-8220. Dostupné z DOI: 10.3390/s16091466.
5. ALAM, Tanweer. A Reliable Communication Framework and Its Use in Internet of Things (IoT). 2018-05, roč. 3.
6. NAIK, N. LPWAN Technologies for IoT Systems: Choice Between Ultra Narrow Band and Spread Spectrum. In: *2018 IEEE International Systems Engineering Symposium (ISSE)*. 2018. Dostupné z DOI: 10.1109/SysEng.2018.8544414.
7. JEBRONI, Zakariae; CHADLI, Hajar; SALMI, Khalid; SABER, Mohammed; TIDHAF, Belkasssem. Modeling and Simulation of LoRaWAN for Smart Metering Network. In: EL MOUSSATI, Ali; KPALMA, Kidiyo; GHAOUTH BELKASMI, Mohammed; SABER, Mohammed; GUÉGAN, Sylvain (ed.). *Advances in Smart Technologies Applications and Case Studies*. Cham: Springer International Publishing, 2020, s. 687–695. ISBN 978-3-030-53187-4.
8. LoRa® and LoRaWAN®: A Technical Overview [online]. [N.d.] [cit. 2021-04-20]. Dostupné z: <https://loro-developers.semtech.com/library/tech-papers-and-guides/loro-and-lorawan/>.
9. PECH, Jiří. IOT TECHNOLOGIE: LORA A LORAWAN. In: [n.d.]. Dostupné také z: <https://www.eman.cz/blog/iot-technologie-lora-a-lorawan-3-5/>.

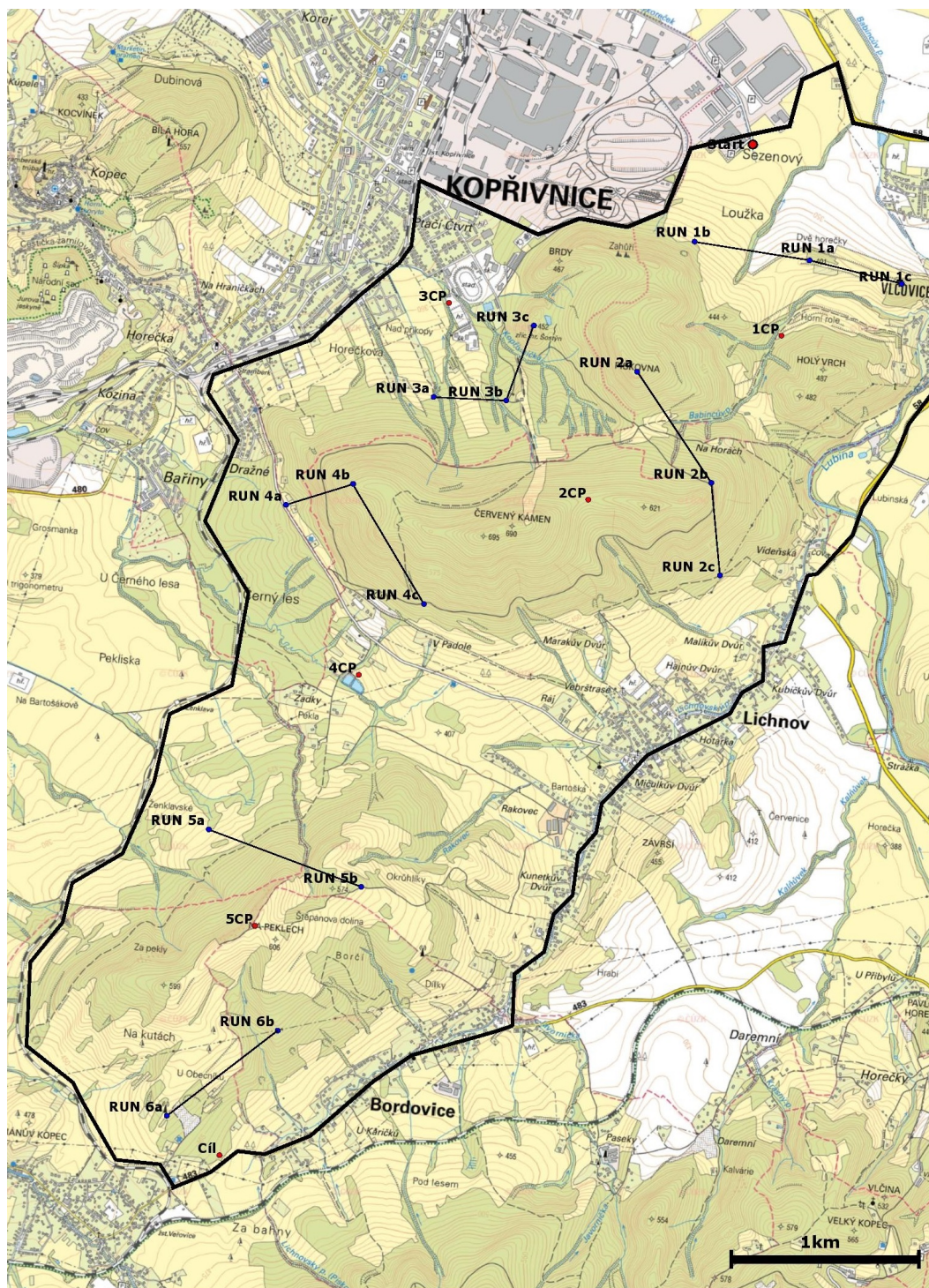
10. JEŘÁBEK, Ondřej. *Signálová analýza LoRa s využitím SDR*. Brno, 2019.
11. BOR, Martin; ROEDIG, Utz. LoRa Transmission Parameter Selection. In: 2017-06. Dostupné z DOI: 10.1109/DCOSS.2017.10.
12. TEAM, Qoitech. How Spreading Factor affects LoRaWAN device battery life [online]. [N.d.] [cit. 2021-01-29]. Dostupné z: <https://www.thethingsnetwork.org/article/how-spreading-factor-affects-lorawan-device-battery-life>.
13. *Technické aspekty technologie Lora* [online] [cit. 2021-01-29]. Dostupné z: <https://pripoj.me/technicke-aspekty-technologie-lora/>.
14. LoRa- (Long Range) Network and Protocol Architecture & Frame Structure [online]. [N.d.] [cit. 2021-01-29]. Dostupné z: <http://www.techplayon.com/lora-long-range-network-architecture-protocol-architecture-and-frame-formats/>.
15. AVOINE, Gildas; FERREIRA, Loïc. Rescuing LoRaWAN 1.0. In: MEIKLEJOHN, Sarah; SAKO, Kazue (ed.). *Financial Cryptography and Data Security*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018. ISBN 978-3-662-58387-6.
16. PETÄJÄJÄRVI, J.; MIKHAYLOV, K.; HÄMÄLÄINEN, M.; IINATTI, J. Evaluation of LoRa LPWAN technology for remote health and wellbeing monitoring. In: *2016 10th International Symposium on Medical Information and Communication Technology (ISMICT)*. 2016-03. ISSN 2326-8301.
17. SILVA, Nuno; MENDES, Jorge; SILVA, Renato; SANTOS, Filipe Neves dos; MESTRE, Pedro; SERÔDIO, Carlos; MORAIS, Raul. Low-Cost IoT LoRa@Solutions for Precision Agriculture Monitoring Practices. In: MOURA OLIVEIRA, Paulo; NOVAIS, Paulo; REIS, Luís Paulo (ed.). *Progress in Artificial Intelligence*. Cham: Springer International Publishing, 2019, s. 224–235. ISBN 978-3-030-30241-2.
18. ROCHA, Lucio A.; BARRETO, Fernando; SEMAN, Laio O. The Internet of Things LoRaWAN Technologies in Academia: A Case Study. In: *The Internet of Things in the Industrial Sector: Security and Device Connectivity, Smart Environments, and Industry 4.0*. Ed. MAHMOOD, Zaigham. Cham: Springer International Publishing, 2019, s. 193–219. ISBN 978-3-030-24892-5. Dostupné z DOI: 10.1007/978-3-030-24892-5_9.
19. *LoRaWAN Specification 1.0.4*. 2020. Dostupné také z: <https://loro-alliance.org/>.
20. NAOUI, Sarra; ELHDHILI, Mohamed; SAIDANE, Leila. Novel Enhanced LoRaWAN Framework for Smart Home Remote Control Security. *Wireless Personal Communications*. 2020-02, roč. 110. Dostupné z DOI: 10.1007/s11277-019-06832-x.
21. *Lokalizace senzorů v síti LoRaWAN*. Praha, 2019.
22. STOKKING, Johan. *The Things Network Architecture*. [N.d.]. Dostupné také z: <https://www.thethingsnetwork.org/article/the-things-network-architecture-1>.

23. What is MQTT? A practical introduction. MQTT – Message Queuing Telemetry Transport [online]. [N.d.] [cit. 2021-02-13]. Dostupné z: <https://www.opc-router.com/what-is-mqtt/>.
24. MUHAMMAD. Shiny Package for R – An Introduction. [N.d.]. Dostupné také z: <http://www.alternatestack.com/development/statistics/r/shiny-package-for-r-an-introduction/>.
25. WICKHAM, Hadley. *Mastering Shiny: Build Interactive Apps, Reports, and Dashboards Powered by R* [online]. 2021. vyd. [N.d.] [cit. 2021-04-21]. ISBN 978-1492047384. Dostupné z: <https://mastering-shiny.org/>.
26. Datasheet ATmega328P: 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash [online]. [N.d.] [cit. 2021-04-24]. Dostupné z: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf.
27. Datasheet RN2483: Low-Power Long Range LoRa Technology Transceiver Module [online]. [N.d.] [cit. 2021-04-24]. Dostupné z: <http://ww1.microchip.com/downloads/en/devicedoc/50002346c.pdf>.
28. Datasheet DS3231: Extremely Accurate I2C-Integrated RTC/TCXO/Crystal [online]. [N.d.] [cit. 2021-04-24]. Dostupné z: <https://datasheets.maximintegrated.com/en/ds/DS3231.pdf>.
29. Datasheet MFRC522: Standard performance MIFARE and NTAG frontend [online]. [N.d.] [cit. 2021-04-24]. Dostupné z: <https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf>.
30. What is RFID? How It Works? Interface RC522 RFID Module with Arduino [online]. [N.d.] [cit. 2021-03-01]. Dostupné z: <https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/>.
31. RN2483 LoRaTECHNOLOGY MODULE COMMAND [online]. [N.d.] [cit. 2021-04-07]. Dostupné z: https://www.quectel.com/UploadImage/Downlad/Quectel_BC95-G%5C&BC68_AT_Commands_Manual_V1.1.pdf.
32. Data Sheet Varta LONGLIFE Power AAA [online]. [N.d.] [cit. 2021-04-13]. Dostupné z: <https://docs.rs-online.com/843b/0900766b81689432.pdf>.

Příloha A

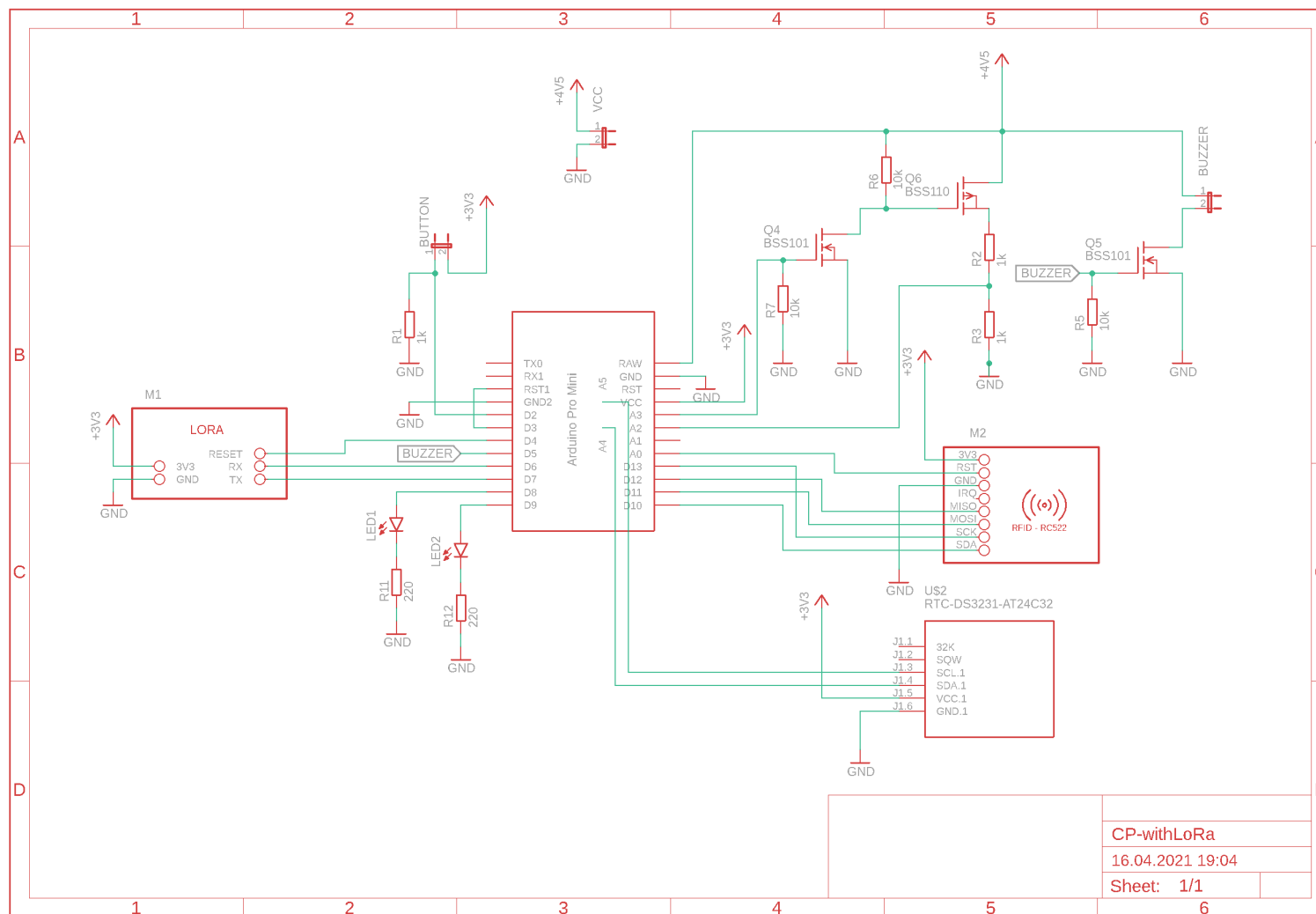
Obrázky

A.1 Ukázka mapy herního území



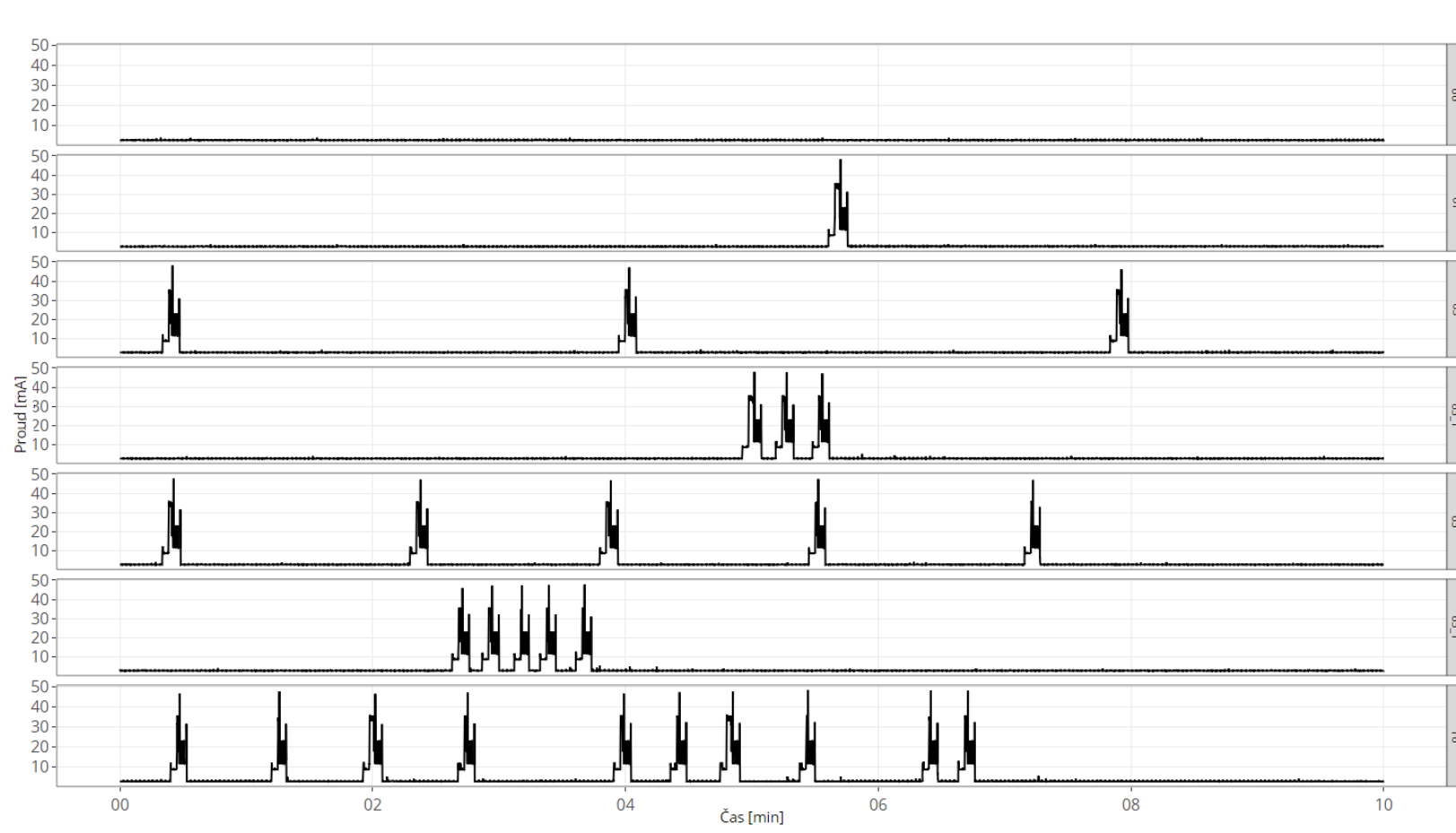
Obrázek A.1: Ukázka mapy herního území

A.2 Schéma zapojení



Obrázek A.2: Schéma zapojení

A.3 Proudový odběr simulace provozu

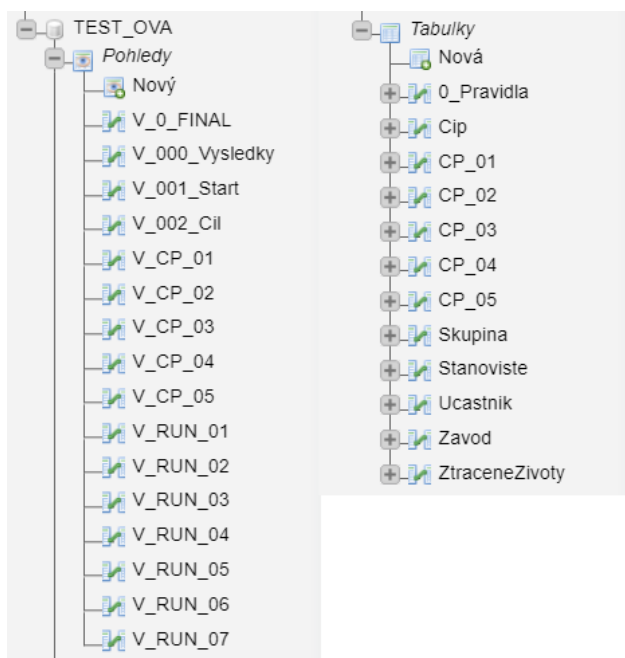


Obrázek A.3: Proudový odběr simulace provozu

Příloha B

Testování

1. Vytvoření nové databáze



Obrázek B.1: Vytvořená databáze (phpMyAdmin)

2. Zápis dat před závodem

Ucastnik

[Import](#) [Odstranit vše](#) [✓ Ukončit editaci](#)

Vybrat soubor
testOva.csv

IMPORT

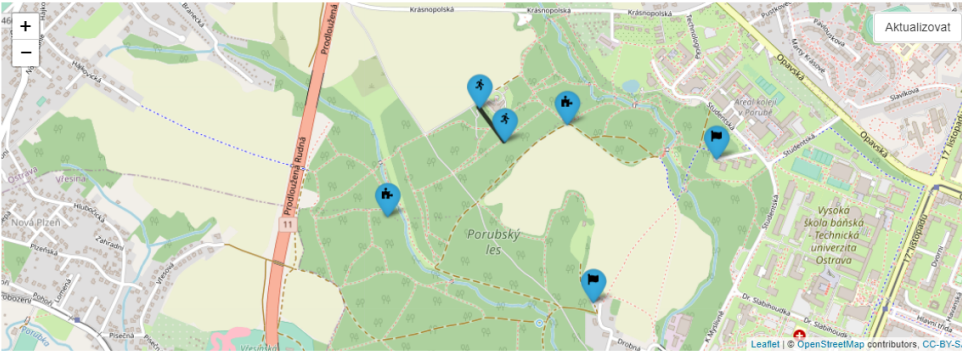
ID	ID_Cip	Jmeno	Prijmeni	Prezdivka	Vyloucen	Akce
1	101	Karel	Mohyla	Kulička	0	✎ ✖
2	102	Radan	Vališ	Lev	0	✎ ✖
3	103	Pavla	Ludvíková	Nika	0	✎ ✖
4	104	Kristián	Váša	Tulák	0	✎ ✖
5	105	Iva	Kutílková	Skřítek	0	✎ ✖

[+ Přidat řádek](#)

Obrázek B.2: Zápis účastníků – import csv

Stanoviste

[Editace](#) [Export do excelu](#)



ID	Nazev	Ukolove	ID_Skupina	Aktivni	GPSN	GPSE	Popis	Poradi
1	START	2		1	49.83567640	18.15579440		1
10	CP_01	1		1	49.83680470	18.14842750	Pamět'	2
11	RUN_01_a	0	1	1	49.83622500	18.14534110	run	3
12	RUN_01_b	0	1	1	49.83731390	18.14411030	run	4
20	CP_02	1		1	49.83386470	18.13952830	Šifry	5
100	CIL	2		1	49.83112610	18.14973170	cil	6

Obrázek B.3: Zápis stanovišť

3. Zápis společného startu

Společný start

<input checked="" type="checkbox"/> Vše	ID	ID_Cíp	Jmeno	Prijmeni	Prezdivka	Vyloucen
<input checked="" type="checkbox"/>	1	101	Karel	Mohyla	Kulička	0
<input checked="" type="checkbox"/>	2	102	Radan	Vališ	Lev	0
<input checked="" type="checkbox"/>	3	103	Pavla	Ludviková	Nika	0
<input checked="" type="checkbox"/>	4	104	Kristián	Váša	Tulák	0
<input checked="" type="checkbox"/>	5	105	Iva	Kutílková	Skřítek	0

Čas: 22.04.2021 15:00 Stanoviště: 1 - START

Obrázek B.4: Společný start – zápis

Zavod

ID	ID_Cíp	ID_Stanoviste	Cas
32	101	1	2021-04-22 15:00:00
33	102	1	2021-04-22 15:00:00
34	103	1	2021-04-22 15:00:00
35	104	1	2021-04-22 15:00:00
36	105	1	2021-04-22 15:00:00

Obrázek B.5: Společný start – zapsané údaje

4. Odstartování závodu

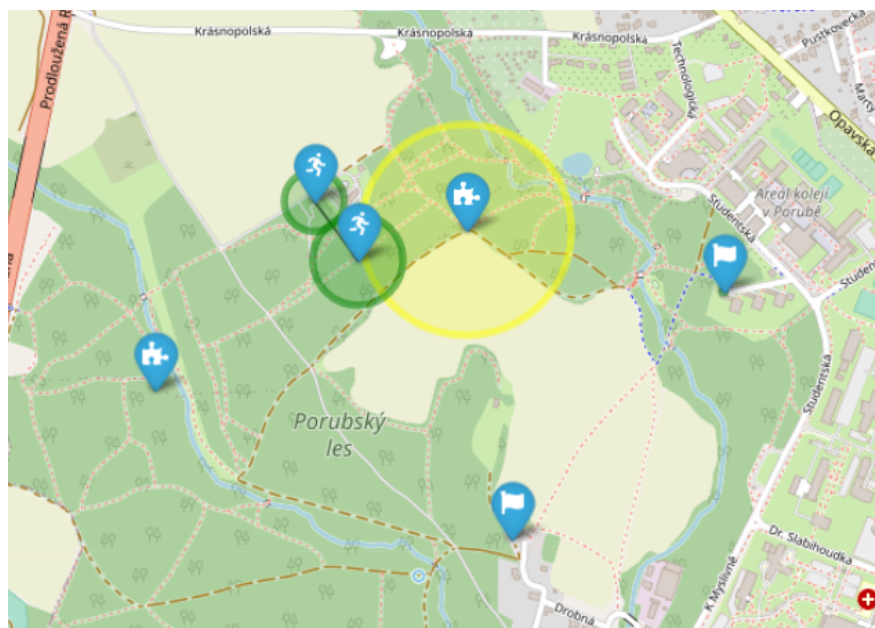
5. Zápis bodů za úkolové stanoviště

CP_02 - Šifry

Editace Export do excelu	
ID_Ucastnik	Body
1	25
2	20
4	13
3	32
5	15

Obrázek B.6: Zápis bodů za úkolové stanoviště – šifry

6. Vizualizace aktuálního stavu závodu



Obrázek B.7: Aktuální stav závodu

7. Doběhnutí účastníků do cíle

8. Vysbírání RFID čipů a průvodek od účastníků

9. Zápis ztracených životů

ZtraceneZivoty

Editace Export do excelu		
ID	ID_Ucastnik	Pocet
1	5	1

Obrázek B.8: Zápis ztracených životů

10. Nahrání dat z RFID čipů na server

Komunikace USB

Připojeno k USB

Data o průchodech

Zápis ID čipuAktualizace času

Po přiložení čipu načte záznamy o průchodech.

☒ Spustit čtení

Možnosti:
☐ Automaticky zapisovat do databáze
☐ Po přečtení smazat

ID čipu	ID stanoviště	Den	Hodiny	Minuty	Sekundy
103	100	22	11	25	51
103	20	22	17	14	44
103	20	22	17	5	47
103	11	22	16	50	43
103	10	22	16	40	2
103	10	22	16	33	50

Obrázek B.9: Nahrání dat z RFID čipů na server – jeden účastník

11. Data z RFID čipů v databázi (phpMyAdmin)

```
SELECT * FROM `Zavod` WHERE `ID_Cip` = 103 ORDER BY `Zavod`.`Cas` ASC
```

☐ Zobrazit vše | Počet řádků: 25 | Filtrovat řádky: Vyhledávání v této tabulce | Se

+ Nastavení

	ID	ID_Cip	ID_Stanoviste	Cas
<input type="checkbox"/> Upravit	34	103	1	2021-04-22 15:00:00
<input type="checkbox"/> Upravit	51	103	10	2021-04-22 16:33:00
<input type="checkbox"/> Upravit	50	103	10	2021-04-22 16:40:00
<input type="checkbox"/> Upravit	49	103	11	2021-04-22 16:52:00
<input type="checkbox"/> Upravit	48	103	20	2021-04-22 17:05:00
<input type="checkbox"/> Upravit	47	103	20	2021-04-22 17:14:00
<input type="checkbox"/> Upravit	46	103	100	2021-04-22 17:25:00

Obrázek B.10: Data z RFID čipů v databázi (phpMyAdmin) – jeden účastník

12. Ověření všech zadaných dat

Detailní výsledky

ID_Ucastnik	Prezdivka	Vyloucen	Zivoty	ZivotyBody	Start	Cil	RUN_01	CP_01	CP_01_T	CP_01_RUN	CP_02	CP_02_T	CP_02_RUN
1	Kulička	0	0	0	15:00:00	17:26:00	20	0	00:00:00	20	25	00:11:00	25
2	Lev	0	0	0	15:00:00	17:19:00	20	20	00:04:00	25	20	00:02:00	25
4	Tulák	0	0	0	15:00:00	17:21:00	20	0	0	0	13	00:07:00	25
3	Nika	0	0	0	15:00:00	17:25:00	20	30	00:07:00	25	32	00:09:00	25
5	Skřítek	0	1	-45	15:00:00	17:20:00	20	0	00:02:00	25	15	00:00:00	20

Obrázek B.11: Detailní výsledky - zvýrazněna chyba v datech

13. Oprava chyby - zápis bodů ze stanoviště

14. Zobrazení výsledkové listiny

Finalní výsledky

Živí

ID_Ucastnik	Prezdivka	ZtraceneZivoty	CasNaTrase	CekaciCas	RUN	CP	RUN_CP	VyslednyCas
3	Nika	0	02:25:00	00:16:00	20	62	50	-00:03:00
2	Lev	0	02:19:00	00:06:00	20	40	50	00:23:00
1	Kulička	0	02:26:00	00:11:00	20	25	45	00:45:00
4	Tulák	0	02:21:00	00:07:00	20	13	25	01:16:00

Mrtví

ID_Ucastnik	Prezdivka	ZtraceneZivoty	CasNaTrase	CekaciCas	RUN	CP	RUN_CP	VyslednyCas
5	Skřítek	1	02:20:00	00:02:00	20	40	45	01:18:00

Ostatní

ID_Ucastnik	Prezdivka	ZtraceneZivoty	CasNaTrase	CekaciCas	RUN	CP	RUN_CP	Vyloucen	VyslednyCas
-------------	-----------	----------------	------------	-----------	-----	----	--------	----------	-------------

Obrázek B.12: Výsledková listina

15. Ukončení závodu

Příloha C

Zdrojové kódy

Součástí práce jsou zdrojové kódy v elektronické podobě. Přílohy jsou přiloženy v IS EDISON.

- Firmware mikrokontroléru
- Webová aplikace
- R Shiny aplikace
- SQL skripty